

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Implémentation d'une méthode d'analyse discriminante sur variables qualitatives

Moons, Xavier

Award date:
1984

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX A NAMUR

INSTITUT D'INFORMATIQUE

IMPLEMENTATION D'UNE METHODE
D'ANALYSE DISCRIMINANTE SUR
VARIABLES QUALITATIVES.

X. MOONS.

Mémoire présenté par

Xavier MOONS

pour l'obtention du grade
de Licencié et Maître en
Informatique.

Année Académique 1983 - 1984.

Je remercie Mme NOIRHOMME - FRAITURE
d'avoir accepté la direction de ce mémoire
et d'avoir guidé mon travail par ses
conseils tout au long de l'année.

TABLE DES MATIERES.

Chapitre 0 : Introduction.

0.1 Présentation du problème de la discrimination et exemples.	0.1
0.2 Présentation des chapitres.	0.5

Chapitre I : Analyse discriminante ordinaire.

I.1 Position du problème et notations.	I.1
I.2 Aspect décisionnel : métrique de discrimination.	I.2
I.2.1 Distance d'un individu à un groupe.	I.2
I.2.2 Choix de la métrique Q.	I.3
I.2.3 Interprétation.	I.3
I.2.4 Méthode de classement.	I.4
I.2.5 Interprétation géométrique dans le cas de deux groupes.	I.5
I.3 Aspect descriptif : axes factoriels discriminants.	I.6
I.3.1 Covariance inter-groupe et intra-groupe.	I.6
I.3.2 Recherche des axes factoriels discriminants.	I.7
I.4 Méthode de "pas-à-pas" pour la sélection de variables.	I.9

Chapitre II : Analyse discriminante sur variables qualitatives :

Méthode de G. Saporta

II.1 Introduction.	II.1
II.2 Sélection progressive d'un sous-ensemble de q variables explicatives.	II.1
II.2.1 Indicatrices associées à une variable X_i à m_i modalités.	II.1
II.2.2 Le coefficient de Tschuprow comme cosinus d'angle.	II.2
II.2.3 Méthode de sélection.	II.3
II.3 Analyse discriminante sur les q variables retenues.	II.5
II.3.1 Analyse factorielle discriminante.	II.5
II.3.2 Procédure de classement.	II.6

Chapitre III : Sélection des variables explicatives.

III.1 Introduction et caractéristiques principales de la sélection.	III.1
III.2 Problèmes et remarques concernant la sélection.	III.2
III.2.1 A propos de la construction du tableau des coefficients de Tschuprow du premier ordre.	III.2
III.2.2 A propos de la transformation du tableau des coefficients de Tschuprow en P2.C.	III.7
III.2.3 A propos du test d'arrêt en P2.B.	III.8

Chapitre IV : Analyse discriminante sur les variables retenues.

IV.1 Analyse factorielle discriminante.	IV.1
IV.1.1 Recherche des facteurs \underline{z}_j .	IV.1
IV.1.2 Analyse factorielle discriminante sur les facteurs \underline{z}_j .	IV.4
IV.1.3 Détermination des codages discriminants.	IV.5
IV.2 Classement des individus.	IV.5

Chapitre V : Problèmes techniques.

V.1 Simple ou double précision.	V.1
V.2 Besoins en place mémoire.	V.1
V.2.1 Principaux tableaux nécessaires pour l'analyse discriminante.	V.1
V.2.2 Tableaux nécessaires pour la sélection des variables discriminantes.	V.3
V.2.3 Solution globale adoptée.	V.3

Chapitre VI : Entrées - sorties.

VI.1 Généralités.	VI.1
VI.2 Données générales nécessaires à l'exécution du premier programme.	VI.2
VI.3 Résultats à attendre de l'exécution du premier programme.	VI.9
VI.4 Données et résultats du deuxième programme.	VI.12

Chapitre VII : Description des programmes.

VII.1 Description du premier programme.	VII.1
VII.1.1 Description de PRINC1.	VII.4
VII.1.2 Description de SAISIE.	VII.10
VII.1.3 Description de NETTOY.	VII.14
VII.1.4 Description de SELECT.	VII.18
VII.1.5 Description de RINTAB.	VII.21
VII.1.6 Description de ROUTAB.	VII.21
VII.1.7 Description de SAVA1, SAVA2, SAVA3, SAVA4, SAVA5, SAVA6, SAVA7, SAVA9, SAVA10, SAVA11, SAVA12, SAVA13 et SAVA14.	VII.21
VII.1.8 Description de ZVLD1, ZVLD2, ZVLD3, ZVLD4, ZVLD5, ZVLD6, ZVLD7, ZVLD8, ZVLD9, ZVLD10, ZVLD11, ZVLD12, ZVLD13 et ZVLD14.	VII.22
VII.1.9 Description de LEC1, LEC2, LEC3, LEC4, LEC5, LEC6, LEC7 et LEC8.	VII.22
VII.1.10 Description de MSGERR et de MSGT.	VII.22
VII.2 Description du deuxième programme.	VII.23
VII.2.1 Description de PRINC2.	VII.25
VII.2.2 Description de SAVA2, SAV43, SAVA4, SAVA5, SAVA6, SAVA10, SAVA13 et SAVA14.	VII.28
VII.2.3 Description de LEC1, LEC2, LEC3, LEC4 et LEC5.	VII.28
VII.2.4 Description de ZVLD1, ZVLD3, ZVLD4, ZVLD5, ZVLD6, ZVLD8, ZVLD10, ZVLD12, ZVLD13, ZVLD14.	VII.28
VII.2.5 Description de MSGERR et MSGT.	VII.28
VII.2.6 Description de ANACOR et ANDISZ.	VII.28

Chapitre VIII : Application.

Annexes.

Annexe A : Le coefficient de Tschuprow comme cosinus d'angle dans
un espace d'opérateurs.

Annexe B : Analyse conjointe de p variables qualitatives.

Annexe C : Mode d'emploi des programmes.

Annexe D : Sources FORTRAN.

Références.

CHAPITRE 0 :

- INTRODUCTION -

0.1 Présentation du problème de la discrimination et exemples.

Soient un ensemble d'individus répartis en plusieurs classes et un ensemble de variables mesurées sur chacun des individus. Le but d'une étude de discrimination est de tenter de répondre à la question suivante : Les variables permettent-elles de différencier les classes considérées?

Cette analyse présuppose l'existence de groupes d'individus bien définis. Ces groupes peuvent être formés dès le départ sur base de considérations naturelles mais peuvent parfois résulter d'une analyse préliminaire. L'analyse discriminante proprement dite ne s'intéresse pas au problème de détermination des groupes. Elle est donc à distinguer d'autres méthodes d'analyse comme les méthodes de classification qui cherchent à mettre en évidence certains groupements qui se dégagent des observations.

Les buts d'une étude de discrimination peuvent être très différents. On distingue en fait deux grands types de problèmes :

- 1) La discrimination à but descriptif
- 2) La discrimination à but décisionnel.

La discrimination à but descriptif.

Etant donné les différentes classes d'individus (pour lesquelles on dispose d'échantillons), on désire mettre en évidence le pouvoir discriminant des variables mesurées sur les individus. On veut, d'une part, s'assurer que l'ensemble des valeurs des variables permette bien de séparer les différentes classes et, d'autre part, on cherchera à com-

parer les valeurs discriminantes des variables et on s'efforcera de déterminer un sous-ensemble de variables permettant la meilleure discrimination entre les classes.

La discrimination à but décisionnel.

C'est ce à quoi on pense en premier lieu lorsqu'on parle de discrimination. Disposant d'un individu pour lequel on connaît les valeurs d'un ensemble de variables mesurées sur lui et sachant que cet individu appartient à une des classes définies à priori et pour lesquelles on a des échantillons de mesures, le problème consiste à affecter cet individu à une des classes.

Voici quelques exemples de problèmes de discrimination qui mettent bien en évidence les différents types de problèmes.

Discrimination à but descriptif :

Exemple 1 [R2].

- - - - -

On dispose de deux échantillons d'individus, les uns normaux, les autres aphasiques. L'ensemble de ces individus a été soumis à un test d'apprentissage d'un code forme-couleur : On leur présente successivement des formes (parmi 6 retenues) et ils doivent alors désigner une couleur (parmi les 6 qu'on leur a indiquées) et l'on note, pour chacune des 6 formes, le nombre de présentations nécessaires pour que le sujet apprenne le code correspondant. Chacun des individus (normaux ou aphasiques) obtient ainsi un vecteur note (à 6 composantes). Les résultats de discrimination obtenus dans ce cas permettent de très bien différencier aphasiques et normaux.

Exemple 2 [R1].

- - - - -

Il s'agit d'expliquer le choix d'un des trois modes de transport suivants : Voiture, train, avion, à l'aide de variables socio-économiques (composition du ménage par tranche d'âge, possession ou non d'une voiture, nombre de voyages effectués dans l'année, ...) et de variables caractérisant le déplacement (région d'arrivée, de départ, distance parcourue, ...). L'étude de discrimination a permis de mettre en évidence trois variables :

- 1) La possession ou non d'un véhicule particulier
- 2) La distance parcourue
- 3) La taille du groupe de voyage.

Discrimination à but décisionnel :

Nous avons vu qu'il s'agit d'affecter un individu à un groupe parmi plusieurs, étant donné que l'on dispose au préalable d'échantillons d'individus pour chacun des groupes. Le fait qu'on ne puisse connaître de façon précise le groupe auquel appartient l'individu alors qu'on le connaît pour tous les individus des échantillons initiaux peut provenir de différentes causes. Citons quelques exemples tirés de Romeder [R2] et qui illustrent chacun une cause différente.

Exemple 3 :

- - - - -

Quel est le sexe d'un individu dont on a découvert le crâne lors de fouilles ?

Pour tenter de répondre à cette question, on dispose de données anthropométriques pour des échantillons que l'on sait provenir d'hommes ou de femmes et l'on effectue les mêmes mesures sur le crâne découvert.

Une autre question serait : Quel est la race d'un individu dont on découvre les ossements et pour lequel on a de bonnes raisons de croire qu'il appar-

tient, par exemple, à une parmi trois races pour lesquelles on dispose d'échantillons d'ossements ?

Il s'agit ici de conjecture sur une information perdue.

Exemple 4 :

- - - - -

Un individu hospitalisé souffre d'ictère et la cause de cet ictère n'a pu être déterminée en première analyse. On ne sait donc pas s'il s'agit d'un ictère chirurgical qui nécessite une opération ou d'un ictère médical qui nécessite, lui, une thérapie médicamenteuse. Cet individu a subi un certain nombre d'examens, soit autant de variables qui permettront de conjecturer s'il s'agit d'un ictère chirurgical ou pas, pourvu que l'on dispose de dossiers comportant les mêmes examens pour les deux échantillons de malades de notre problème.

Il s'agit ici d'une information inaccessible.

Exemple 5 : Reconnaissance de formes :

- - - - -

Si on se limite aux lettres minuscules manuscrites, on dispose de 26 classes. Pour caractériser chacune de ces classes, on peut procéder ainsi : On quadrille le carré dans lequel se trouve la lettre en un certain nombre N de petits carrés. La plupart de ces petits carrés seront blancs tandis que certains d'entre eux auront un peu de noir à l'intérieur. On attribue une valeur comprise entre 0 et n , et proportionnelle à l'intensité du noir dans chaque petit carré. On caractérise ainsi une lettre écrite par une personne par un vecteur de R^N .

Si l'on dispose pour chacune des lettres de l'alphabet d'un échantillon de représentations différentes (i.e. : la même lettre écrite par des individus différents), on peut essayer de reconnaître des lettres écrites par d'autres personnes.

Il s'agit ici, d'information accessible à l'homme mais non à la machine.

Exemple 6 :

- - - - -

Peut-on prévoir un résultat post-opératoire à partir d'un certain nombre de données caractérisant l'état du malade avant l'intervention ? Ce problème qui intéresse beaucoup les médecins provient du fait que l'information est inexistante.

Il en est de même de la conjecture de la réussite d'un étudiant sur base de ses résultats antérieurs et du milieu social dont il est issu.

Le champ d'application de l'analyse discriminante est, comme on peut le constater, très étendu et on n'en finirait plus de citer des exemples, que ce soit en médecine, anthropologie, biologie, économie, etc...

0.2 Présentation des chapitres .

=====

Le premier chapitre est un rappel des éléments de base de la discrimination dans le cas de variables quantitatives. On y traite en premier lieu l'aspect décisionnel qui comporte le choix de la métrique qui paraît la mieux adaptée au problème et le classement des individus. Vient ensuite l'aspect descriptif avec la recherche des axes factoriels discriminants. La dernière section est un rappel du principe des méthodes de sélection d'un sous-ensemble de variables.

Le deuxième chapitre est un exposé très bref de la méthode de Saporta qui permet de traiter des variables qualitatives. La première partie se rapporte à une généralisation des procédures de "pas-à-pas" pour la sélection de variables qualitatives avec comme critères à maximiser, des coefficients de Tschuprow partiels. La deuxième explique dans les grandes lignes la discrimination sur les facteurs z : comment obtenir ceux-ci et comment en déduire les axes factoriels discriminants. On montre

également que le classement des individus est beaucoup plus simple du fait que ceux-ci sont décrits par les facteurs z.

Le troisième chapitre est une analyse détaillée de la méthode de sélection des variables explicatives et de tous les problèmes qui peuvent se poser. On en déduit des contraintes sur les données et on propose quelques méthodes pour les traiter.

Le quatrième chapitre se rapporte à la partie analyse discriminante proprement dite de la méthode de Saporta, c'est-à-dire, ce qui se passe après la sélection. On y trouve la recherche des facteurs z, l'élimination de certains d'entre eux, l'analyse factorielle discriminante sur ceux qui restent, la détermination des codages discriminants et le classement des individus.

Le cinquième chapitre expose quelques problèmes d'implémentation (place mémoire, simple ou double précision) et les choix qui en ont résulté , tandis que le sixième traite des décisions qui ont été prises pour les entrées-sorties.

Le septième chapitre comporte la description des deux programmes tandis que le huitième consiste en l'exposé d'une application de la méthode en biologie : il s'agit de déterminer parmi des tests aux antibiotiques quels sont ceux qui permettent de discriminer au mieux entre différents groupes de microbes.

CHAPITRE I :

- ANALYSE DISCRIMINANTE ORDINAIRE -

I.1 Position du problème et notations.

=====

On dispose d'un ensemble de N individus répartis en g groupes définis à priori par les modalités d'une variable \mathcal{Y} , et d'un ensemble de p autres variables, toutes quantitatives, mesurées sur chaque individu. Le tableau X des données est de la forme:

	1	...	j	...	p
1					
:					
:					
i			x_{ij}		
:					
:					
N					

où x_{ij} représente la modalité de la variable X_j prise par l'individu i . On voit donc que chaque individu peut être représenté par un point de R^p dont les p composantes sont les modalités que les variables prennent sur cet individu, tandis que la donnée d'une variable est équivalente à la donnée d'un point de R^N , dont les N composantes sont les différentes modalités de cette variable prises par les individus. On éliminera du tableau de données les colonnes correspondant à des variables constantes et, si chaque individu x_i est doté d'un poids p_i tel que $\sum_{i=1}^N p_i = 1$, on munira R^N de la métrique $D_p = \text{diag} (p_i)$.

Deux questions, liées chacune à un aspect de l'analyse discriminante, peuvent alors se poser :

- aspect décisionnel : Dans quel groupe classer un individu supplémentaire ?
- aspect descriptif : Ces p variables permettent-elles de différencier les groupes ?

Etudions ces deux aspects en détail.

I.2 Aspect décisionnel : métrique de discrimination.

=====

I.2.1 Distance d'un individu à un groupe.

Disposant d'un individu supplémentaire, c'est-à-dire un vecteur \underline{a} de R^p dont la $i^{\text{ième}}$ composante représente la modalité de la $i^{\text{ième}}$ variable prise par cet individu, on désire l'affecter à un des g groupes (on supposera au départ qu'il appartient effectivement à un de ces groupes). La règle de décision étant d'affecter l'individu au groupe auquel il est le plus semblable, ou encore, duquel il est le plus proche, il convient de déterminer ce que nous entendons par distance entre un individu et un groupe d'individus.

Soient $\underline{a} \in R^p$ un individu et $x_i^k \in R^p$, $i = 1, \dots, N_k$ les individus du $k^{\text{ième}}$ groupe G_k , N_k étant l'effectif de ce groupe. La distance de \underline{a} au $i^{\text{ième}}$ individu du groupe sera donnée par

$$d^2(\underline{a}, x_i^k) = (\underline{a} - x_i^k)^t \cdot Q \cdot (\underline{a} - x_i^k),$$

où Q est une métrique sur R^p , i.e. une matrice $p \times p$, symétrique et définie positive. Pour la distance de \underline{a} au groupe G_k , nous prendrons la racine carrée positive de la moyenne des carrés des distances de \underline{a} à chaque individu du groupe :

$$D^2(\underline{a}, G_k) = \frac{1}{N_k} \cdot \sum_{i=1}^{N_k} (\underline{a} - x_i^k)^t \cdot Q \cdot (\underline{a} - x_i^k).$$

Le résultat de l'affectation dépendra donc du choix de la métrique Q . Suivant la démarche de Romeder [R2, p. 19 - 31], nous déterminons au paragraphe suivant, une métrique qui semble bien adaptée au problème de la discrimination.

I.2.2 Choix de la métrique Q.

Nous choisirons une métrique Q (avec la contrainte $|Q| = 1$ où $|Q|$ désigne le déterminant de Q) qui minimise le carré moyen de la distance entre individus d'un même groupe. Soit x_i^k , un individu du groupe G_k . Sa distance au groupe G_k^x formé des $N_k - 1$ individus restants est donnée par

$$D^2(x_i^k, G_k^x) = \frac{1}{N_k - 1} \cdot \sum_{j=1}^{N_k} d^2(x_i^k, x_j^k).$$

La distance moyenne au carré sera donc

$$D_k^2 = \frac{1}{N_k \cdot (N_k - 1)} \cdot \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} d^2(x_i^k, x_j^k)$$

et le problème

$$\begin{aligned} &\text{Min } D_k^2 \\ &\quad Q \\ &\text{s.c. } |Q| = 1 \end{aligned}$$

aura pour solution [R2, p. 27-31]

$$D_k^2 = \frac{2 \cdot N_k \cdot p}{N_k - 1} \cdot |S_k|^{\frac{1}{p}}$$

avec

$$Q_k = |S_k|^{\frac{1}{p}} \cdot S_k^{-1}$$

où S_k est la matrice de variance-covariance des variables pour le groupe considéré.

I.2.3 Interprétation.

La distance d'un individu \underline{a} à un groupe G_k

$$D^2(\underline{a}, G_k) = \frac{|S_k|^{1/p}}{N_k} \cdot \sum_{i=1}^{N_k} (\underline{a} - x_i^k)^t \cdot S_k^{-1} \cdot (\underline{a} - x_i^k)$$

peut également s'écrire [R3, p. 58]

$$D^2(\underline{a}, G_k) = |S_k|^{\frac{1}{p}} \cdot [p + (\underline{a} - \bar{x}^k)^t \cdot S_k^{-1} \cdot (\underline{a} - \bar{x}^k)]$$

où \bar{x}^k est le vecteur dont la $i^{\text{ième}}$ composante est la moyenne de la $i^{\text{ième}}$ variable sur les N_k individus du groupe. On voit ainsi que les surfaces d'équidistance à un groupe G_k sont des ellipsoïdes centrés en \bar{x}^k et de matrice S_k^{-1} .

Pour $S_k = \text{diag} (s_i^{k^2})$, $i = 1, \dots, p$, la distance est donnée par

$$D^2(\underline{a}, G_k) = \left(\prod_{i=1}^p s_i^{k^2} \right) \cdot \left[p + \sum_{i=1}^p \left(\frac{a_i - \bar{x}_i^k}{s_i} \right)^2 \right].$$

Ceci montre bien que Q est une métrique qui met en évidence l'agrégation du groupe. En effet, une variable de faible variance (grande concentration) se verra affectée d'un poids important.

I.2.4 Méthode de classement.

Nous affecterons l'individu supplémentaire \underline{a} au groupe G_{k_0} qui vérifie

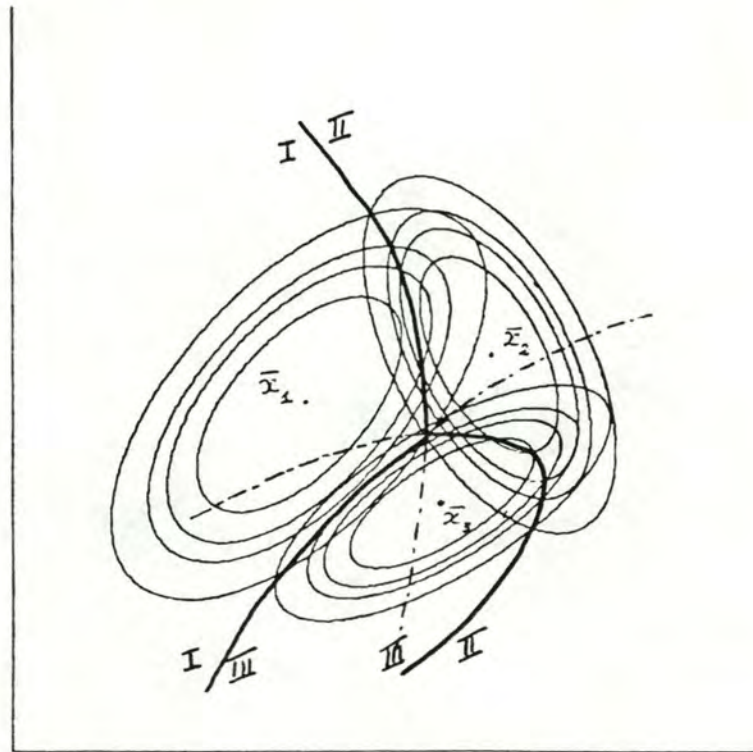
$$D^2(\underline{a}, G_{k_0}) = \min [D^2(\underline{a}, G_k); k = 1, \dots, g]$$

avec

$$D^2(\underline{a}, G_k) = |S_k|^{\frac{1}{p}} \cdot [p + (\underline{a} - \bar{x}^k)^t \cdot S_k^{-1} \cdot (\underline{a} - \bar{x}^k)].$$

Puisque les courbes d'équidistances sont des ellipsoïdes centrés en \bar{x}^k et de matrice S_k^{-1} , les courbes d'indifférence entre deux groupes sont des surfaces du second degré. Ces surfaces permettent de délimiter les régions correspondant aux décisions de classement.

Reprenant l'exemple de Bragard [R3, p. 67] pour $g = 3$ et $p = 2$, nous obtenons graphiquement :



Vu que la métrique est différente pour chaque groupe, la méthode d'affectation nécessite le calcul du déterminant et de l'inverse de toutes les matrices S_k . C'est pourquoi on se contente souvent d'adopter pour distance entre \underline{a} et G_k pour tout k , l'expression

$$D^2(\underline{a}, G_k) = |S|^{-\frac{1}{p}} \cdot [p + (\underline{a} - \bar{x}^k)^t \cdot S^{-1} \cdot (\underline{a} - \bar{x}^k)]$$

où S est la matrice de variance-covariance totale, ce qui revient à considérer que les dispersions sont homogènes dans les g groupes. Il suffit alors de comparer les quantités $(\underline{a} - \bar{x}^k)^t \cdot S^{-1} \cdot (\underline{a} - \bar{x}^k)$ pour $k = 1, \dots, g$. Le terme $\underline{a}^t \cdot S^{-1} \cdot \underline{a}$ étant commun, il reste à comparer $(\bar{x}^k - 2\underline{a})^t \cdot S^{-1} \cdot \bar{x}^k$.

I.2.5 Interprétation géométrique dans le cas de deux groupes.

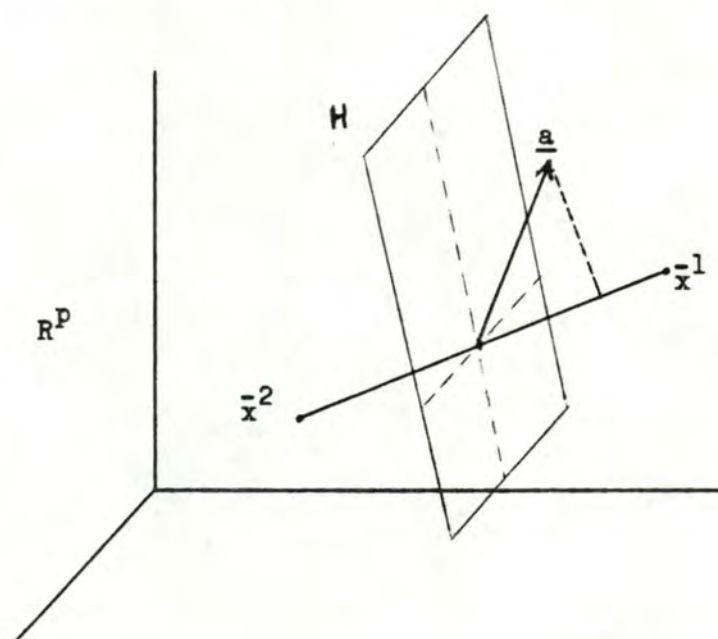
Dans le cas de deux groupes G_1 et G_2 , il suffit de comparer les quantités $(\bar{x}^1 - 2\underline{a})^t \cdot S^{-1} \cdot \bar{x}^1$ et $(\bar{x}^2 - 2\underline{a})^t \cdot S^{-1} \cdot \bar{x}^2$ et la règle de décision

est d'affecter \underline{a} au groupe G_1 si $(\bar{x}^1 - 2\underline{a})^t \cdot S^{-1} \cdot \bar{x}^1 < (\bar{x}^2 - 2\underline{a})^t \cdot S^{-1} \cdot \bar{x}^2$.

Développant cette condition, nous obtenons

$$(\bar{x}^1 - \bar{x}^2)^t \cdot S^{-1} \cdot [\underline{a} - \frac{1}{2}(\bar{x}^1 + \bar{x}^2)] > 0,$$

ce qui signifie que, si la S^{-1} - projection du vecteur $\underline{a} - \frac{1}{2}(\bar{x}^1 + \bar{x}^2)$ joignant le point milieu de $\bar{x}^1 \bar{x}^2$ au point \underline{a} de R^p , sur l'axe $\bar{x}^2 - \bar{x}^1$ est positive, on affecte l'individu \underline{a} au groupe G_1 . La partition de R^p en deux régions de décision est donc définie par un hyperplan H.



I.3 Aspect descriptif : axes factoriels discriminants.

=====

I.3.1 Covariance inter-groupe et covariance intra-groupe.

Soit $S = (s_{ij})$ $i, j = 1, \dots, p$ la matrice de variance-covariance totale

$$s_{ij} = \sum_{k=1}^N p_k \cdot (x_{ki} - g_i) \cdot (x_{kj} - g_j).$$

$g_i = \sum_{k=1}^N p_k \cdot x_{ki}$ est la moyenne de la variable X_i sur les N individus.

Si nous nous limitons aux individus d'un groupe G_k , la matrice de variance-covariance associée sera $S^k = (s_{ij}^k)$ $i, j = 1, \dots, p$ avec

$$s_{ij}^k = \sum_{h=1}^{N_k} p_h \cdot (x_{hi} - g_i^k) \cdot (x_{hj} - g_j^k)$$

où

$$g_i^k = \frac{1}{P(G_k)} \cdot \sum_{h=1}^{N_k} p_h \cdot x_{hi}$$

est le centre de gravité des individus du groupe G_k pour la variable X_i , $P(G_k) = \sum_{h=1}^{N_k} p_h$ étant le poids du groupe. $W = \sum_{k=1}^G S^k$ s'appellera "matrice de covariance intra-groupe".

On montre facilement que $S = W + B$ avec $B = (b_{ij})$ $i, j = 1, \dots, p$ et

$$b_{ij} = \sum_{k=1}^G P(G_k) \cdot (g_i^k - g_i) \cdot (g_j^k - g_j).$$

B s'appelle "matrice de covariance inter-groupe" car elle représente la dispersion des centres de gravité des groupes par rapport au centre de gravité total, chaque terme de la somme étant pondéré par le poids du groupe.

I.3.2 Recherche des axes factoriels discriminants.

Nous recherchons un ensemble de nouvelles variables, combinaisons linéaires des anciennes, qui permettent de discriminer au mieux entre les groupes. Ceci revient à déterminer des vecteurs de R^p tels que, après projection sur l'espace engendré par ceux-ci, les groupes soient le mieux séparés, c'est-à-dire, tels que la dispersion inter-groupe soit maximale et celle intra-groupe minimale.

Comme premier axe, nous prendrons donc $c_1 \in R^p$ qui maximise le rapport $c_1^t \cdot B \cdot c_1 / c_1^t \cdot W \cdot c_1$, ce qui est équivalent à maximiser $c_1^t \cdot B \cdot c_1$

sous la contrainte $c_1^t . S . c_1 = 1$. La solution à ce problème est donnée par le vecteur propre de $S^{-1} . B$ correspondant à la plus grande valeur propre f_1 [R4, p. 12]. Le deuxième axe, c_2 , qui maximise $c_2^t . B . c_2$ sous la contrainte $c_2^t . S . c_2 = 1$ et qui est S - orthogonal au premier, sera le vecteur propre de $S^{-1} . B$ correspondant à la deuxième valeur propre (si on classe celles-ci par ordre décroissant). ...

Le $k^{ième}$ axe c_k qui maximise $c_k^t . B . c_k$ sous la contrainte $c_k^t . S . c_k = 1$ et qui est S - orthogonal à tous les précédents est le vecteur propre de $S^{-1} . B$ associé à la $k^{ième}$ valeur propre f_k .

Les f_i , qui représentent en fait les covariances inter-groupe après projection sur les c_i , s'appellent " pouvoir discriminant " des différents " facteurs discriminants " $X . c_i$. Nous aurons $0 \leq f_i \leq 1$ et f_i ne sera égal à 1 que si la covariance intra-groupe est nulle, c'est-à-dire, si pour chaque groupe, les projections des points du groupe coïncident avec la projection du centre de gravité de ce groupe.

Le nombre d'axes factoriels discriminants sera égal au nombre de valeurs propres non nulles de $S^{-1} . B$ ou, en d'autres termes, au rang de cette matrice. Il sera donc inférieur ou égal à $\min [p, g-1]$ qui, dans la pratique, vaudra souvent $g-1$.

Lorsque $g = 2$, il n'existe qu'un seul axe discriminant \underline{c} . En représentant les projections sur \underline{c} des individus des deux groupes par des signes différents, on obtient une représentation de la discrimination entre les groupes. Si g est supérieur à 2, on peut également la représenter en projetant les différents groupes sur le plan déterminé par les deux premiers axes factoriels discriminants (c'est la meilleure représentation plane discriminante).

I.4 Méthode de " pas-à-pas " pour la sélection de variables.

Plus le nombre de variables est grand, plus le coût de traitement est élevé, que ce soit pour expliquer le pouvoir discriminant des variables ou pour classer des individus. C'est pourquoi on cherche à déterminer un sous-ensemble de q variables semblant discriminer le mieux. La méthode pour sélectionner un des C_p^q sous-ensembles de q variables est une méthode itérative qui fonctionne de la manière suivante :

- Au pas 1, on évalue un critère sur les p variables et on choisit comme première variable celle pour laquelle il est maximal.

.....

- Au pas k , $k > 1$, on évalue un critère sur les $p-k+1$ variables restantes conjointement aux $k-1$ déjà sélectionnées, et on choisit comme $k^{i\text{ème}}$ variable, celle pour laquelle il est maximal.

Remarquons que le critère varie d'un pas à l'autre car il doit tenir compte des variables sélectionnées aux pas précédents et dont le choix n'est pas remis en question.

Plusieurs critères sont utilisables comme, par exemple, " le pourcentage de bien-classés " ou " la trace de $S^{q-1} \cdot B^q$ " [R2] et certains fournissent un critère d'arrêt à la sélection. Comme ces critères ne sont pas directement applicables dans le cas de variables qualitatives, je n'en dirai pas plus.

CHAPITRE II :

- ANALYSE DISCRIMINANTE SUR VARIABLES
QUALITATIVES : METHODE DE G. SAPORTA -

II.1 Introduction.

=====

On dispose de N individus répartis en g groupes définis par les modalités d'une variable \mathcal{Y} et on veut "expliquer" cette partition et classer des individus ultérieurs grâce à p variables qualitatives $\mathcal{X}_1, \dots, \mathcal{X}_p$ à m_1, \dots, m_p modalités. Si on remplace chaque \mathcal{X}_i par l'ensemble des m_i variables indicatrices de ses modalités, on est ramené à un problème d'analyse discriminante usuelle avec $m_1 + \dots + m_p$ variables numériques explicatives. Le problème de discrimination s'identifie alors à la recherche d'un codage optimal des p variables.

Deux difficultés apparaissent :

- La première provient du fait que l'ensemble des variables indicatrices n'est pas de rang plein; il existe donc une infinité de codages conduisant au même optimum.
- La seconde apparaît lorsque l'on veut adapter les méthodes de sélection progressive des variables couramment utilisées en analyse discriminante ordinaire. On ne peut en effet introduire une indicatrice seule sans celles qui correspondent à la même variable qualitative. Il faudra donc traiter les indicatrices par bloc.

II.2 Sélection progressive d'un sous-ensemble de q variables explicatives.

=====

II.2.1 Indicatrices associées à une variable \mathcal{X}_i à m_i modalités.

Soit \mathcal{X}_i une variable qualitative à m_i modalités, mesurée sur N individus. Cette variable qualitative est équivalente à un tableau X_i de m_i indicatrices $x_1^i, \dots, x_{m_i}^i$ où $x_j^i(k) = 1$ si l'individu k prend la modalité j de la variable \mathcal{X}_i , et 0 sinon.

$$X_i \equiv \begin{array}{c} x_1^i \ x_2^i \ \dots \ x_{m_i}^i \\ \begin{array}{cccc} 1 & 0 & 1 & \dots & 0 \\ 2 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ N & 0 & 0 & \dots & 0 \end{array} \end{array} \equiv (x_{kj}^i) \begin{array}{l} k=1, \dots, N \\ j=1, \dots, m_i \end{array}$$

Ce tableau d'indicatrices engendre une variété linéaire de R^N de dimension m_i , $W_i = \{ \underline{w} \mid \underline{w} = X_i \cdot \underline{a} ; \underline{a} \in R^{m_i} \}$, qui est en fait le sous-espace de tous les codages possibles pour χ_i ; \underline{a} étant le vecteur des codes.

Remarque 1 : Chaque W_i contient le vecteur $\underline{1}_N$ de R^N (vecteur dont toutes les composantes sont égales à 1).

Remarque 2 : Si nous centrons les \underline{w} , ils sont D_p - orthogonaux au vecteur $\underline{1}_N$ et les W_i correspondants sont de dimension $m_i - 1$.

II.2.2 Le coefficient de Tschuprow comme cosinus d'angle.

Pour mesurer l'interdépendance de deux variables qualitatives χ_a et χ_b , on peut utiliser le coefficient de K. Pearson

$$PHI_{ab}^2 = -1 + \sum_{i=1}^{m_a} \sum_{j=1}^{m_b} \frac{d_{ij}^2}{d_{i.} \cdot d_{.j}}$$

où $d_{ij} = \sum_{k=1}^N p_k \cdot x_{ki}^a \cdot x_{kj}^b$ est la somme des poids des individus prenant en même temps la $i^{ième}$ modalité de la variable χ_a et la $j^{ième}$ modalité de la variable χ_b . C'est l'élément (i,j) de la matrice $X_a^t \cdot D_p \cdot X_b$.

$d_{i.} = \sum_{j=1}^{m_b} d_{ij} = \sum_{k=1}^N p_k \cdot x_{ki}^a$ est le poids de la $i^{ième}$ modalité de la variable χ_a tandis que $d_{.j}$ est le poids de la $j^{ième}$ modalité de la variable χ_b .

Pour obtenir une mesure comprise entre 0 et 1 qui ne dépende pas du nombre de modalités des variables et pour pouvoir de ce fait comparer les inter-relations entre différentes variables, on utilise le coefficient de Tschuprow

$$T_{ab}^2 = \frac{\text{PHI}_{ab}^2}{\sqrt{(m_a-1).(m_b-1)}} .$$

Il vaut 1 si $m_a = m_b$ et si le tableau croisé $X_a^t . D_p X_b$ peut se mettre sous la forme diagonale par simple permutation des colonnes. C'est le cas de l'association complète.

Le coefficient de Tschuprow peut s'interpréter en terme de cosinus d'angle dans un espace d'opérateurs de R^N dans R^N . [R5], [annexe A]
En effet, $T_{ij}^2 = \cos(P_{i0}, P_{j0})$ où P_{i0} et P_{j0} sont les projecteurs D_p -orthogonaux sur les sous-espaces W_{i0} et W_{j0} des variables centrées codant χ_i et χ_j , le cosinus employé étant celui associé au produit scalaire de la trace du produit : $\langle A; B \rangle = \text{tr}(A.B)$. Si on identifie χ_i au projecteur P_{i0} , ce cosinus est donc l'analogie d'un coefficient de corrélation entre variables qualitatives.

II.2.3 Méthode de sélection.

La première variable sélectionnée χ_1 est celle qui maximise le coefficient de Tschuprow avec γ . Pour sélectionner les autres variables, on définit des coefficients de Tschuprow partiels qui tiennent compte des variables déjà retenues en se basant sur la formule de trigonométrie suivante :

$$\cos \delta = \frac{\cos \beta - \cos \alpha . \cos \gamma}{\sin \alpha . \sin \gamma} \quad [R6, p. 370]$$

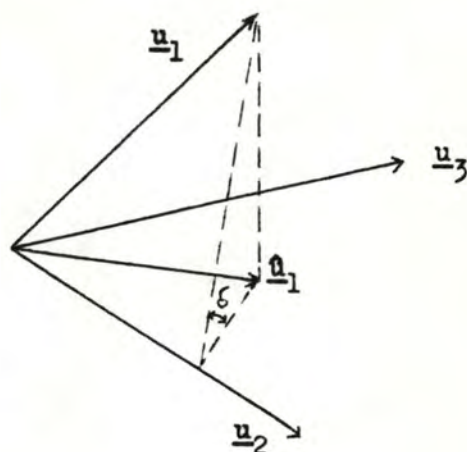
avec $\alpha = (\underline{u}_1, \underline{u}_2)$

$\beta = (\underline{u}_1, \underline{u}_3)$

$\gamma = (\underline{u}_2, \underline{u}_3)$

$\delta = (\underline{u}_1, \underline{a}_1)$

\underline{a}_1 = projection de \underline{u}_1 sur
le plan déterminé par
 \underline{u}_2 et \underline{u}_3 .



On définit des coefficients de Tschuprow partiels entre les variables y et χ_i relativement à χ_1 par

$$r_{y\chi_i/\chi_1} = \frac{r_{y\chi_i} - r_{y\chi_1} \cdot r_{\chi_1\chi_i}}{\sqrt{(1-r_{y\chi_1}^2) \cdot (1-r_{\chi_1\chi_i}^2)}} \quad (x)$$

La deuxième variable sélectionnée χ_2 sera celle qui maximise $|r_{y\chi_i/\chi_1}|$.

Elle a pour propriété d'être très liée à y tout en l'étant le moins possible à χ_1 . Pour la suite, à chaque étape, les variables sélectionnées aux étapes précédentes ne sont pas remises en cause, et on ajoute comme nouvelle variable, celle qui maximise un coefficient de Tschuprow d'ordre supérieur obtenu en itérant la formule (x). Le processus s'arrête lorsque l'on a sélectionné un nombre de variables fixé à l'avance ou lorsque le coefficient de Tschuprow multiple défini par

$$r_{y;\chi_1 \dots \chi_q} = \sqrt{1 - (1-r_{y\chi_1}^2) \cdot \dots \cdot (1-r_{y\chi_q/\chi_1 \dots \chi_{q-1}}^2)}$$

n'augmente plus suffisamment.

II.3 Analyse discriminante sur les q variables retenues.

II.3.1 Analyse factorielle discriminante.

Après l'étape de sélection, on dispose de q variables χ_1, \dots, χ_q et donc d'un tableau $X = (X_1^t \dots X_q^t)$ y correspondant. L'analyse factorielle discriminante consiste à rechercher les vecteurs propres de $S^{-1}.B$ où S est la matrice de covariance totale et B la matrice de covariance inter-groupe. S et B auront pour expression, si nous supposons que X est centré, ce que nous pouvons faire sans perdre de généralité,

$$S = X^t . D_p . X$$

$$B = X^t . D_p . Y . (Y^t . D_p . Y)^{-1} . Y^t . D_p . X$$

où Y est le tableau des indicatrices associées à γ .

Comme le rang de X est au plus égal à $m_1 + m_2 + \dots + m_q - q = m - q$, S n'est pas inversible. Pour tourner cette difficulté, on remplace X par une base de W, l'espace engendré par X, ce qui revient à choisir une pseudo-inverse à $X^t . D_p . X$. Le fait de remplacer X par cette base est équivalent à imposer une contrainte aux codages des variables. Saporta montre en effet [R5, p. III.12] que choisir l'inverse de Moore-Penrose est équivalent au choix de codages centrés pour les χ_i .

Une méthode pour obtenir cette pseudo-inverse, consiste à remplacer X par la matrice Z dont les $m-q$ colonnes \underline{z}_j sont les facteurs de l'analyse conjointe des q variables [annexe B]. Les \underline{z}_j vérifient

$$\underline{z}_j = X . \underline{u}_j$$

avec
$$D^{-1} . X^t . D_p . X . \underline{u}_j = f_j . \underline{u}_j$$

où D est la diagonale de $X^t . D_p . X$.

Comme les \underline{z}_j sont centrés, non corrélés et que l'on peut les choisir de variance unité, la matrice de covariance totale pour les \underline{z}_j sera la matrice unité et l'analyse factorielle discriminante pour ces nouvelles variables en sera fortement simplifiée car il suffira alors de rechercher les vecteurs propres de

$$B_Z = Z^t \cdot D_p \cdot Y \cdot (Y^t \cdot D_p \cdot Y)^{-1} \cdot Y^t \cdot D_p \cdot Z.$$

Le $k^{\text{ième}}$ facteur discriminant pour Z est donné par

$$\underline{d}^k = Z \cdot \underline{c}^k$$

avec

$$B_Z \cdot \underline{c}^k = \varepsilon_k \cdot \underline{c}^k,$$

\underline{c}^k étant le $k^{\text{ième}}$ axe factoriel discriminant ou codage discriminant pour Z. Pour obtenir les codages discriminants \underline{a}^j de l'analyse de X, il suffit de comparer \underline{d}^j à $X \cdot \underline{a}^j$. On obtient directement

$$\underline{a}^j = \sum_{k=1}^{m-g} c_k^j \cdot \underline{u}_k$$

et donc, le $j^{\text{ième}}$ codage discriminant s'obtient par combinaison linéaire des codages descriptifs avec pour coefficients, ceux du $j^{\text{ième}}$ codage discriminant pour les facteurs \underline{z} .

II.3.2 Procédure de classement.

L'affectation d'un individu \underline{a} à l'un des g groupes se fait selon la procédure usuelle présentée en I.2.4, la métrique utilisée étant $|\underline{S}|^{1/g} \cdot \underline{S}^{-1}$. Vu que les individus sont décrits par les facteurs \underline{z} , que pour ceux-ci la matrice \underline{S} est la matrice unité et que, de ce fait, les calculs sont moins coûteux, il est intéressant d'exprimer \underline{a} par ses coordonnées dans la base des \underline{z}_j avant d'effectuer l'affectation.

Si \underline{e} est le vecteur de description logique de l'individu

$$\underline{e} = (\overbrace{0100}^{m_1} \overbrace{00100}^{m_2} \dots \overbrace{10001}^{m_g})^t,$$

ses coordonnées dans la base des \underline{z}_j sont données par $U^t \cdot \underline{e}$, où U est la matrice dont les colonnes sont les \underline{u}_j , et les quantités à comparer

$$\bar{x}^k \cdot (X^t \cdot D_p \cdot X)^{-1} \cdot \bar{x}^k - 2 \cdot \bar{x}^k \cdot (X^t \cdot D_p \cdot X)^{-1} \cdot \underline{e}$$

deviennent après changement de base

$$\bar{z}^k \cdot \bar{z}^k - 2 \cdot \bar{z}^k \cdot U^t \cdot \underline{e}.$$

L'individu sera donc affecté au groupe k_0 qui vérifie

$$\bar{z}_0^k \cdot \bar{z}_0^k - 2 \cdot \bar{z}_0^k \cdot U^t \cdot \underline{e} = \min \left\{ \bar{z}^k \cdot \bar{z}^k - 2 \cdot \bar{z}^k \cdot U^t \cdot \underline{e}; k \in \{1, \dots, g\} \right\}.$$

CHAPITRE III :

- SELECTION DES VARIABLES EXPLICATIVES -

III.1 Introduction et caractéristiques principales de la sélection.

=====

Comme nous l'avons déjà mentionné, la méthode de G. Saporta est une méthode de " pas-à-pas " avec comme critère de sélection à l'étape k , la maximisation du coefficient de Tschuprow d'ordre k . Plus précisément, elle se résume de la manière suivante :

P1 : Initialisation de la sélection.

Elle comporte essentiellement la formation du tableau des coefficients de Tschuprow du premier ordre entre les variables explicatives et y ainsi que entre les variables explicatives elles-mêmes.

P2 : A chaque étape k , $k \geq 1$, on effectue :

P2.A Sélection de la $k^{\text{ième}}$ variable en prenant parmi celles qui n'ont pas encore été sélectionnées, celle qui maximise le coefficient de Tschuprow d'ordre k avec y . C'est-à-dire

$$\chi_{i_k} \text{ telle que } |T_{i_k,0}^{(k)}| = \max \{ |T_{i,0}^{(k)}| ; \chi_i \text{ non déjà sélectionnée} \}$$

P2.B " Test d'arrêt ".

Il peut s'effectuer de deux manières :

P2.B.1 L'utilisateur a fixé au départ le nombre NVS de variables à sélectionner et il suffit alors de tester si le nombre de variables déjà sélectionnées a atteint ou n'a pas atteint NVS.

P2.B.2 On calcule le coefficient de Tschuprow multiple d'ordre k défini par

$$T_m^{(k)} = \sqrt{1 - (1 - T_{i_1,0}^{(1)})^2 \cdot \dots \cdot (1 - T_{i_k,0}^{(k)})^2}$$

ou encore, en fonction de celui d'ordre $k-1$

$$T_m^{(k)} = \sqrt{1 - (1 - T_m^{(k-1)})^2 \cdot (1 - T_{i_k,0}^{(k)})^2}.$$

L'utilisateur a fixé une quantité QTM au départ, et on arrête lorsque $T_m^{(k-1)} + QTM > T_m^{(k)}$; C'est-à-dire lorsque l'on estime que le coefficient de Tschuprow multiple n'augmente plus suffisamment.

P2.C S'il n'y a pas arrêt, construction du tableau des coefficients de Tschuprow d'ordre $k+1$ en fonction de ceux d'ordre k et de la dernière variable sélectionnée, par la relation

$$T_{i,j}^{(k+1)} = \frac{T_{i,j}^{(k)} - T_{i,i_k}^{(k)} \cdot T_{i_k,j}^{(k)}}{\sqrt{(1 - T_{i,i_k}^{(k)})^2 \cdot (1 - T_{i_k,j}^{(k)})^2}}$$

où $T_{i,j}^{(k)}$ représente l'élément (i,j) de ce tableau au pas k , c'est-à-dire le coefficient de Tschuprow d'ordre k entre les variables χ_i et χ_j , et où i_k est l'indice de la variable sélectionnée au pas k . Il ne reste plus qu'à recommencer à partir de P2.A avec ce nouveau tableau.

P3 : S'il y a arrêt, " clôture de la sélection ".

Elle consiste principalement à arranger les résultats en ne gardant que ce qui concerne les variables sélectionnées et γ , sur lesquelles s'effectuera l'analyse factorielle.

Analysons à présent les différentes étapes de la sélection.

III.2 Problèmes et remarques concernant la sélection.

=====

III.2.1 A propos de la construction du tableau des coefficients de Tschuprow du premier ordre.

L'élément (i,j) , $T_{i,j}^{(1)}$, qui est le coefficient de Tschuprow

d'ordre 1 entre les variables χ_i et χ_j , est défini par

$$\tau_{i,j}^{(1)} = \frac{-1 + \sum_{k=1}^{m_i} \sum_{h=1}^{m_j} \frac{d_{kh}^2}{d_{k\cdot} \cdot d_{\cdot h}}}{\sqrt{(m_i-1) \cdot (m_j-1)}} \quad (\text{voir } \S \text{ II.2.2})$$

Problème 2.1.1 : $d_{k\cdot} = 0$ ($d_{\cdot h} = 0$) pour un certain indice k (h).

Cela veut dire que la $k^{\text{ième}}$ modalité de χ_i ($h^{\text{ième}}$ modalité de χ_j) n'est pas prise; elle a un poids nul et peut donc être éliminée avant la sélection.

Il est également souhaitable de supprimer les modalités ayant un poids très faible pour ne pas augmenter inutilement le volume des calculs et l'imprécision des résultats. On éliminera donc toutes les modalités de poids inférieur à une certaine quantité PMMIN, $0 < \text{PMMIN} < 1$.

Malheureusement, cette suppression crée un nouveau problème :

Problème 2.1.2 : Un individu prend une modalité supprimée.

Si un individu prend la modalité j de la variable χ_k et si cette modalité est supprimée, cela provoque la formation d'un "trou" dans le tableau des codages condensés. Comment procéder pour le combler ?

Une première solution est de supprimer cet individu du tableau des données. Ceci n'est raisonnable que si le nombre d'observations est suffisamment élevé pour que la perte d'information engendrée par la suppression n'influence pas trop l'analyse.

Une deuxième solution est de dire que l'on n'a pas pu mesurer la variable χ_k sur cet individu et, de ce fait, lui attribuer la modalité "non-répondu" pour cette variable. Je pense que cette manière d'agir est à proscrire car elle risque de donner une importance beaucoup trop grande à une modalité qui ne nous apprend rien.

Une troisième solution consiste à attribuer à l'individu en question une modalité j de la variable X_k prise au hasard parmi les modalités conservées de la variable. Cette solution convient très bien lorsque les modalités n'ont pas de rapport entre elles mais, lorsqu'elles sont liées, il vaut mieux employer la méthode suivante.

La quatrième solution est de grouper plusieurs modalités se ressemblant. Il me semble raisonnable de laisser le choix à l'utilisateur entre les solutions un, trois et quatre.

Remarques sur les différentes méthodes proposées.

1) Solution par suppression des individus.

C'est la méthode qui pose le plus de problèmes car la suppression du ou des individus se rapportant à une modalité de poids trop faible peut provoquer des modifications en cascade. En effet, chaque fois qu'un individu est supprimé, les poids des modalités prises par l'individu pour les différentes variables décroissent du poids de cet individu, ce qui a pour conséquence que, certaines modalités qui étaient de poids suffisant avant le début du nettoyage se retrouvent avec un poids trop faible. Ceci nous oblige à reparcourir l'ensemble des modalités pour s'assurer que toutes celles qui sont de poids non nul, sont de poids suffisant et cela, chaque fois que des individus ont été supprimés.

Inversement, la somme des poids des individus devant toujours valoir 1, Il est nécessaire de répartir, soit proportionnellement, soit uniformément, la somme des poids des individus supprimés sur les individus restants. Les poids des modalités vont augmenter et certaines modalités qui étaient de poids trop faible avant la suppression, seront de poids suffisant après.

2) Solution par ventilation sur les modalités conservées.

Cette méthode est de loin la plus simple mais elle ne peut pas être employée s'il existe une variable dont toutes les modalités sont de

poids trop faible.

3) Solution par regroupement de modalités.

Cette méthode nécessite que l'utilisateur donne pour chaque variable et pour chaque modalité de cette variable, la modalité "groupe", c'est-à-dire, la nouvelle modalité en cas de poids trop faible.

Exemple : Si pour une variable, on a les modalités 1, 2, 3, 4, 5 et 6, on pourra avoir pour les modalités groupes, 2, 2, 2, 4, 5 et 4, ce qui aura pour effet de :

- grouper les modalités 1, 2 et 3 sous le numéro 2 si l'une des trois est de poids non nul mais trop faible.
- grouper les modalités 4 et 6 sous le numéro 4 si l'une des deux est de poids non nul mais trop faible.
- ne rien changer pour la modalité 5.

Les numéros des modalités groupes sont soumis à certaines contraintes :

- Le numéro d'une modalité groupe d'une variable sera toujours le numéro d'une des modalités de cette variable.
- Si le numéro d'une modalité est celui d'une modalité groupe pour d'autres modalités de la variable, il l'est également pour cette modalité.

Il faut aussi être certain que, après regroupement, les poids sont suffisants.

Solution mixte : suppression-groupement.

Dans certains cas, il est intéressant d'employer la suppression pour des modalités ne concernant que peu d'individus; dans d'autres, il est préférable de grouper les modalités et, en général, l'utilisateur ne sait pas dire à l'avance quelle méthode il vaut mieux employer et, si c'est la méthode par regroupement, comment grouper les modalités pour que les modalités groupes soient de poids suffisant. Il doit donc lui être possible d'effectuer ce choix en cours d'exécution sur base des poids et des nombres

d'individus concernés par les différentes modalités de la variable dont on traite une modalité de poids trop faible. L'utilisateur aura la possibilité, soit de supprimer les individus concernés par cette modalité, soit de la jumeler à une autre de poids non nul mais pas obligatoirement de poids suffisant.

Remarque 2.1.3 :

- - - - -

La deuxième solution page III.3 met en évidence le problème que représente un poids trop important pour la modalité "non-répondu" d'une variable. Si ce poids dépasse un seuil $PNRMAX$, $0 \leq PNRMAX < 1$, fixé par l'utilisateur, il faudra éliminer cette variable. Comme les modalités sont identifiées par des numéros, il faut prendre une convention concernant la place de la modalité "non-répondu" pour une variable afin de pouvoir tester son poids. J'ai choisi de la placer en dernière position dans la liste des modalités d'une variable.

Problème 2.1.4 : $m_i = 1$ ou $m_j = 1$.

- - - - -

Cela signifie que la variable χ_i (χ_j) est constante sur toute la population. Elle n'apporte rien au problème de la discrimination et peut donc être éliminée avant la sélection.

Remarque 2.1.5 :

- - - - -

$0 \leq T_{i,j}^{(1)} \leq 1$ et $T_{i,j}^{(1)}$ ne peut valoir 1 que si $m_i = m_j$ et si $X_i^t \cdot D_p \cdot X_j$ peut se mettre sous forme diagonale par simple permutation des lignes ou des colonnes. Cela veut dire que deux individus prennent une même modalité de χ_i si et seulement si ils prennent une même modalité de χ_j . Les variables χ_i et χ_j sont parfaitement redondantes au point de vue de la discrimination et on peut en laisser tomber une des deux. Généralisant ce raisonne-

ment lorsque $T_{i,j}^{(1)}$ est proche de 1, l'utilisateur déterminera un seuil SR, $0 < SR < 1$ à partir duquel on considérera que deux variables sont redondantes et, si $SR \leq T_{i,j}^{(1)} \leq 1$, on éliminera une des deux variables, le choix de celle-ci étant quelconque.

III.2.2 A propos de la transformation du tableau des coefficients de Tschuprow en P2.C.

Un problème peut se poser lors de l'application de la formule de transformation lorsqu'une variable a un coefficient de Tschuprow d'ordre k proche de 1 avec la variable sélectionnée; plus exactement, lorsque $|T_{i,i_k}^{(k)}| \simeq 1$ ou $|T_{j,i_k}^{(k)}| \simeq 1$. Considérons les différentes possibilités :

1) $|T_{j,i_k}^{(k)}| \simeq 1$. Cela veut dire que la variable sélectionnée à l'étape k, ajoutée à celles qui ont été choisies aux étapes précédentes, permet d'expliquer complètement \mathcal{Y}_j . Il ne sera donc pas nécessaire de pousser la sélection plus loin.

2) $|T_{i,i_k}^{(k)}| \simeq 1$ pour un i compris entre 1 et le nombre de variables explicatives.

- Si $k=1$, ce problème a été éliminé par la remarque 2.1.5

- Si $k>1$, de façon analogue au point 1), cela signifie que la variable \mathcal{X}_i n'apporte rien de plus à la discrimination que l'ensemble des variables déjà sélectionnées et peut, de ce fait, être éliminée.

Remarque 2.2.1 :

- - - - -

Comme n'interviennent dans la formule de transformation que des coefficients entre variables non encore sélectionnées et \mathcal{Y}_j , il n'est pas nécessaire de mettre à jour les lignes et les colonnes du tableau des coefficients correspondant à des variables éliminées ou déjà sélectionnées.

III.2.3 A propos du test d'arrêt en P2.B.

Que l'on emploie la méthode du coefficient de Tschuprow multiple ou celle pour laquelle le nombre de variables à sélectionner est fixé à l'avance, la sélection peut très bien ne pas aboutir faute de variable sélectionnable. En effet, un certain nombre de variables sont susceptibles d'être éliminées, que ce soit pour nombre insuffisant de modalités, poids trop important de la modalité "non-répondu", coefficient de Tschuprow d'ordre 1 avec une autre variable supérieur à SR ou encore, pour cause de coefficient de Tschuprow d'ordre k avec la variable χ_{i_k} , supérieur au seuil de redondance fixé.

Si l'utilisateur fixe lui-même le nombre NVS de variables à sélectionner, la seule contrainte que l'on puisse imposer est que NVS soit supérieur ou égal à 1 et inférieur ou égal à NV, le nombre de variables au départ. Remarquons que dans ce cas encore, le nombre de variables sélectionnées peut ne pas atteindre NVS si on se trouve trop tôt dans la situation où le coefficient de Tschuprow d'ordre k entre la variable sélectionnée χ_{i_k} et la variable y_j est supérieur ou égal en valeur absolue au seuil SR.

Si l'utilisateur veut employer la seconde méthode, il devra prendre certaines précautions quant au choix de QTM. Comme le coefficient de Tschuprow multiple est croissant, il faudra que QTM soit strictement supérieur à 0 pour que cette méthode ait un sens. De plus, pour que le nombre de variables sélectionnées soit supérieur ou égal à 1, il faudra que QTM soit inférieur ou égal à $T_m^{(1)}$ et, comme au départ, l'utilisateur n'a aucune idée de ce que pourra valoir $T_m^{(1)}$, si QTM lui est supérieur, on conservera la première variable sélectionnée. Dans les autres cas, $k > 1$, si QTM est supérieur à $T_m^{(k)}$, on ne retiendra pas la variable sélectionnée au pas k .

CHAPITRE IV :

- ANALYSE DISCRIMINANTE SUR LES VARIABLES RETENUES -

IV.1 Analyse factorielle discriminante.

Rappels concernant le tableau de données.

Le tableau de données X consiste en la juxtaposition de q tableaux logiques X_j où, dans chaque ligne i de X_j , on trouve $m_j - 1$ fois zéro et une fois 1 dans la colonne correspondant à la modalité prise par l'individu n° i . La somme des vecteurs colonnes de X_j est donc toujours égale à $\underline{1}$; la somme des vecteurs colonnes de X valant $q \cdot \underline{1}$. La somme des éléments d'une colonne est égale à l'effectif de la modalité correspondante.

Le nombre de lignes est le même pour chaque X_j tandis que le nombre de colonnes varie. Si m_j désigne le nombre de colonnes de X_j , $m = \sum_{j=1}^q m_j$ sera le nombre de colonnes de X .

IV.1.1 Recherche des facteurs \underline{z}_j .

Comme je l'ai déjà mentionné, les X_j étant des tableaux d'indicateurs, on ne peut inverser $X^t \cdot D_p \cdot X$ qui est de rang au plus égal à $m - q + 1$ ($m - q$ si on se borne aux codages centrés). C'est pourquoi on remplace le tableau X des variables explicatives par le tableau Z des facteurs \underline{z}_j obtenus en effectuant l'analyse conjointe des q variables qualitatives [annexe B], ce qui donne exactement les mêmes résultats que si on avait effectué l'analyse des correspondances de X . Les facteurs \underline{z}_j vérifient :

$$\underline{z}_j = X \cdot \underline{u}_j$$

avec \underline{u}_j vecteur propre de $M = D^{-1} \cdot X^t \cdot D_p \cdot X$.

A propos des éléments propres de M .

Bien que la matrice M ne soit pas symétrique, ses valeurs propres sont toutes réelles. En effet, il est évident que M a les mêmes valeurs propres que la matrice $P = \sum_{i=1}^q P_i$ où P_i est la matrice de projection D_p -ortho-

gonale sur le sous-espace W_i engendré par X_i . Les vecteurs propres de P sont les facteurs \underline{z}_j recherchés. P étant une somme de matrices D_p -symétriques est elle-même D_p -symétrique. L'avantage d'analyser M plutôt que P réside dans le fait que M est une matrice $m \times m$ tandis que P est de dimension $N \times N$ et, si le nombre des observations augmente, on arrive très vite à des impossibilités pratiques.

Tous les sous-espaces W_i ayant en commun le vecteur $\underline{1}_N$, $\underline{1}_m$ est vecteur propre trivial de M associé à la valeur propre q .

La dimension de $\bigoplus_{i=1}^q W_i$ étant au plus égale à $m-(q-1)$, cela entraîne que 0 est valeur propre de M avec une multiplicité au moins égale à $q-1$.

M possède donc $m-q$ valeurs propres non triviales. Leur somme vaut $m-q$.

De plus, toutes les valeurs propres sont supérieures ou égales à zéro et inférieures ou égales à q .

Comme M est une matrice D et $X^t \cdot D_p \cdot X$ symétrique, ses vecteurs propres sont D et $X^t \cdot D_p \cdot X$ orthogonaux et, de ce fait, les facteurs \underline{z}_j associés sont D_p -orthogonaux deux-à-deux. Il est également évident que les facteurs \underline{z}_j non triviaux sont centrés puisque D_p -orthogonaux au facteur trivial $\underline{1}_N$.

A propos de la recherche des éléments propres de M .

Le problème posé par le fait que M n'est pas nécessairement symétrique n'est pas important car D^{-1} est diagonale et définie positive et, de ce fait, rechercher les éléments propres de $D^{-1} \cdot X^t \cdot D_p \cdot X$ est équivalent à rechercher ceux de $D^{-\frac{1}{2}} \cdot X^t \cdot D_p \cdot X \cdot D^{-\frac{1}{2}}$ qui est symétrique. En effet, ces deux matrices ont mêmes valeurs propres et \underline{v} est vecteur propre de $D^{-\frac{1}{2}} \cdot X^t \cdot D_p \cdot X \cdot D^{-\frac{1}{2}}$ si et seulement si $\underline{u} = D^{-\frac{1}{2}} \cdot \underline{v}$ est vecteur propre de M . Il suffira donc de rechercher les valeurs propres et les vecteurs propres de $D^{-\frac{1}{2}} \cdot X^t \cdot D_p \cdot X \cdot D^{-\frac{1}{2}}$, de multiplier les composantes de ces derniers par les inverses des racines carrées des poids des modalités et le problème sera réglé.

Il existe beaucoup de méthodes pour rechercher les valeurs propres et les vecteurs propres d'une matrice symétrique et la plupart d'entre elles sont déjà implémentées, entre autres, dans la librairie EISPACK [R8]. Toutes ces procédures ont coutume de normer les vecteurs propres trouvés à 1 pour la métrique usuelle $\text{diag}(1, \dots, 1)$; les vecteurs propres de M sont alors D - normés à 1. Remarquons que, si toutes les valeurs propres sont distinctes, les vecteurs propres sont uniques (à un facteur multiplicatif près) et le vecteur $\underline{1}_m$ est effectivement donné comme vecteur propre associé à la plus grande valeur propre q ; mais, si q est de multiplicité supérieure à 1, le choix des vecteurs propres associés est arbitraire et, en général, ces routines ne prennent pas le vecteur $\underline{1}_m$ dans le système orthonormé qu'elles choisissent. Il faut donc effectuer une rotation de tous les vecteurs propres de manière à amener un de ceux associés à la plus grande valeur propre (on peut prendre comme convention le dernier) sur le vecteur $\underline{1}_m$.

Soit \underline{x} , un vecteur quelconque. Pour pouvoir l'amener au moyen d'une rotation sur un autre vecteur \underline{y} , il faut évidemment qu'ils soient de même norme. Or, dans notre cas, $\underline{y} (\underline{1}_m)$ est D - normé à \sqrt{q} tandis que \underline{x} , lui, est D - normé à 1. La première chose à faire consiste donc à D - normer $\underline{1}_m$ à 1 en divisant ses composantes par \sqrt{q} . La matrice de rotation R telle que $R.\underline{x} = \underline{y}$ est alors de la forme $I - 2.w.w^t.D$ avec

$$w = \frac{\underline{y} - \underline{x}}{\|\underline{y} - \underline{x}\|_D}$$

et elle ne nécessite la mémorisation que du seul vecteur w .

Les vecteurs transformés seront alors les vecteurs propres de la matrice $R.D^{-1}.X^t.D_p.X.R^{-1}$ qui est exactement égale à $D^{-1}.X^t.D_p.X$.

IV.1.2 Analyse factorielle discriminante sur les facteurs \underline{z}_j .

Comme les \underline{z}_j sont centrés, cette analyse consiste en la recherche des valeurs propres g_k et des vecteurs propres \underline{c}^k de la matrice

$$B_Z = (Z^t \cdot D_p \cdot Z)^{-1} \cdot Z^t \cdot D_p \cdot Y \cdot (Y^t \cdot D_p \cdot Y)^{-1} \cdot Y^t \cdot D_p \cdot Z.$$

Etant donné que les \underline{z}_j sont D_p - orthogonaux, il y a intérêt à les D_p - normer à 1 car ainsi l'expression de B_Z est simplifiée.

$$B_Z = Z^t \cdot D_p \cdot Y \cdot (Y^t \cdot D_p \cdot Y)^{-1} \cdot Y^t \cdot D_p \cdot Z$$

Pour ce faire, on peut agir, soit directement sur les \underline{z}_j , soit par l'intermédiaire des \underline{u}_j en tenant compte des relations suivantes :

$$\|\underline{z}_j\|_{D_p}^2 = \underline{z}_j^t \cdot D_p \cdot \underline{z}_j = f_j \cdot \|\underline{u}_j\|_D^2 = \|\underline{u}_j\|_{X^t D_p X}^2.$$

Il suffit donc de $X^t D_p X$ - normer \underline{u}_j à 1 ou de le D - normer à $1/\sqrt{f_j}$ pour que \underline{z}_j soit D_p - normé à 1.

Elimination de certains facteurs.

En pratique, il est inutile de garder les $m-q$ facteurs \underline{z}_j et on peut effectuer la discrimination sur une partie d'entre eux. En effet, les \underline{z}_j sont des variables les plus liées à l'ensemble des X_i au sens où elles maximisent la somme des carrés des coefficients de corrélation empirique avec les X_i et les valeurs maximales de ces sommes sont justement les f_j . Afin d'éviter de discriminer sur du bruit, on éliminera donc les \underline{z}_j de trop faible inertie. Il suffira de rechercher les valeurs propres f_j de M se trouvant dans l'intervalle $[f_{j\min}, q]$ où $f_{j\min}$ est une valeur fixée par l'utilisateur et vérifiant $0 < f_{j\min} \leq q$.

De plus, il est inutile de conserver les facteurs \underline{z}_j dont le pouvoir discriminant est trop faible. Rappelons que les pouvoirs discriminants des \underline{z}_j , qui sont en fait les variances inter-groupes sont donnés par

$$\underline{z}_j^t \cdot D_p \cdot Y \cdot (Y^t \cdot D_p \cdot Y)^{-1} \cdot Y^t \cdot D_p \cdot \underline{z}_j$$

et ces pouvoirs sont additifs car les \underline{z}_j sont non corrélés.

Après avoir reclassé les \underline{z}_j restants de la première suppression par ordre de pouvoir discriminant décroissant, on ne gardera que ceux dont le pouvoir discriminant cumulé est supérieur à une certaine borne PDISCU fixée par l'utilisateur et vérifiant $0 < \text{PDISCU} < \text{NGR}-1$ où NGR est le nombre de groupes d'individus (on les gardera tous si la somme des pouvoirs discriminants n'atteint pas PDISCU).

IV.1.3 Détermination des codages discriminants.

L'évaluation des codages discriminants ne pose aucun problème.

Il suffit de calculer pour chaque vecteur propre \underline{c}^j de B_Z la somme

$$\underline{a}^j = \sum_{k=1}^{m^0} c_k^j \cdot \underline{u}_k$$

qui s'écrit matriciellement $\underline{a}^j = U^0 \cdot \underline{c}^j$ où U^0 est la matrice dont les colonnes sont les \underline{u}_j correspondant aux \underline{z}_j conservés.

IV.2 Classement des individus.

=====

Le classement des individus est également très simple si on dispose de U et de \bar{z}^k , $k = 1, \dots, g$. Or nous avons U (les vecteurs propres de M) et \bar{z}^k est donné par la $k^{\text{ième}}$ ligne de la matrice

$$(Y^t \cdot D_p \cdot Y)^{-1} \cdot Y^t \cdot D_p \cdot X \cdot U$$

dont le calcul ne pose aucun problème.

Pour se donner une idée de la qualité de la discrimination obtenue, il peut être intéressant de reclasser, outre les individus supplémentaires, les individus qui ont servis de base à l'analyse. On indiquera

donc, pour chaque groupe, le nombre d'individus de ce groupe affectés dans chaque groupe. Remarquons qu'il faut être très prudent quant aux conclusions à tirer de ce tableau. En effet, si x pourcents des individus du groupe n° i sont reclassés dans le groupe n° j , cela ne voudra pas dire que, prenant un individu supplémentaire, il a une probabilité x d'être classé dans le groupe n° j alors qu'il appartient au groupe n° i . Il ne faut jamais perdre de vue que ce tableau de reclassement, ainsi que la procédure de classement sont tous les deux construits à partir des individus de départ.

CHAPITRE V :

- PROBLEMES TECHNIQUES -

V.1 Simple ou double précision.

=====

Le langage utilisé étant le FORTRAN 77, une question se pose d'emblée : Est-il nécessaire d'employer la double précision avec éventuellement les problèmes de place mémoire qu'elle provoque, ou peut-on se contenter de la simple précision ?

La réponse à cette question est très simple étant donné que, en général, on peut s'attendre à une erreur de l'ordre de $\epsilon.n^2$ à $\epsilon.n^3$ sur le résultat d'une recherche d'éléments propres d'une matrice symétrique (ϵ étant la précision de la machine et n l'ordre de la matrice) et sachant que les matrices à analyser peuvent être au maximum de dimension 150×150 (c'est une borne que je me suis fixée). Si on emploie la simple précision, la précision à laquelle on pourra s'attendre pour les valeurs propres et les vecteurs propres issus de la première analyse sera beaucoup trop faible (de l'ordre de 10^{-2} à 10^{-4}) d'autant plus que ces résultats sont l'objet de nombreux traitements par la suite. La double précision s'impose donc.

V.2 Besoins en place mémoire.

=====

Lorsqu'on écrit un programme, on est toujours confronté au problème de minimiser deux choses "contradictoires" : la place mémoire et le temps d'exécution. J'ai malgré tout essayé de minimiser le temps calcul tout en occupant une place mémoire raisonnable mais qui permette tout de même de traiter des applications de taille respectable.

V.2.1 Principaux tableaux nécessaires pour l'analyse discriminante.

La recherche des valeurs propres de $D^{-\frac{1}{2}} X^t D_p X D^{-\frac{1}{2}}$ dans l'intervalle $[f_{j,\min}, q]$ et des vecteurs propres associés nécessite principalement deux

gros tableaux : A de dimension $m \times m$ qui contient la matrice à analyser et B de dimension $m \times (m-q+1)$ qui contient les vecteurs propres \underline{u}_j résultants de l'analyse.

La formation de la matrice $Y^t D_p Z$ à partir de la matrice $Y^t D_p X$ ne prend pas plus de place que cette dernière, c'est-à-dire un tableau de dimension $NGR \times m$.

Le calcul des pouvoirs discriminants et la transformation des tableaux en fonction des facteurs conservés ne nécessite pratiquement rien de plus.

Un petit vecteur suffit pour stocker la matrice $Y^t D_p Y$ et la matrice B_Z peut prendre la place de $D^{-\frac{1}{2}} X^t D_p X D^{-\frac{1}{2}}$ qui ne sert plus à rien.

Comme il faut toutes les valeurs propres et tous les vecteurs propres de B_Z , j'emploie une procédure qui stocke les vecteurs propres à la place de la matrice de départ. L'analyse discriminante sur les facteurs \underline{z} ne nécessite donc pratiquement pas de place supplémentaire.

Le calcul des codages discriminants peut se faire directement à partir de ce qui se trouve dans A et B .

Le reste des calculs concernant le classement des individus se fait également sans extension importante.

En résumé, l'analyse discriminante nécessite trois tableaux importants :

$$A (m \times m)$$

$$B (m \times (m-q+1))$$

$$C (NGR \times m)$$

plus quelques vecteurs de dimension au plus m .

Remarque :

- - - - -

J'aurais pu n'employer que A et C en me servant pour l'analyse des correspondances de la même procédure que pour l'analyse discriminante sur les facteurs \underline{z} , mais cela aurait diminué la vitesse d'exécution (calcul de plus de valeurs propres et de plus de vecteurs propres, sauvetage de la

matrice U des vecteurs propres dans un fichier avant la formation de la matrice B_Z , complications pour le calcul des codages discriminants, restitution de U en mémoire pour le classement des individus).

V.2.2 Tableaux nécessaires pour la sélection des variables discriminantes.

Les principaux tableaux utilisés sont : $X^t_D X$, $Y^t_D X$ et le tableau des coefficients de Tschuprow T entre les variables. Ils ont respectivement $m \times m$, $NGR \times m$ et $p \times p$ composantes. Comme $X^t_D X$ et T sont symétriques, il ne faudra en retenir que $m \times (m+1)/2$, $NGR \times m$ et $p \times (p+1)/2$. Si nous nous limitons à 9 groupes d'individus, ce qui paraît tout à fait raisonnable, et à 8 modalités au maximum pour les variables (plus une pour la modalité "non-répondu"), on se rend compte que, étant donné que toutes les composantes de ces tableaux occupent deux mots mémoire, on ne pourra guère dépasser 30 variables sans sortir de l'espace virtuel principal dont on dispose (128 K), ce qui est trop peu; on ne peut imposer au départ que le nombre total de modalités soit inférieur ou égal à 270. Dépasser les 128 K impliquant un ralentissement sensible de la vitesse d'exécution, j'ai décidé de calculer le tableau de Burt par tranches en me servant d'un fichier intermédiaire, ce qui ne nécessite plus que $9 \times m$ mots double précision sans trop augmenter le temps d'exécution. Cette façon de procéder me permet de traiter un nombre plus que suffisant de variables.

V.2.3 Solution globale adoptée.

Si la limitation due à la place mémoire est insignifiante pour la sélection des variables, ce n'est plus le cas lors de l'analyse discriminante où on ne peut pratiquement plus dépasser 150 modalités, ce qui correspond à avoir gardé après la sélection une trentaine de variables ayant en moyenne 5 modalités. Cette limite, inacceptable au départ est, ici,

tout à fait raisonnable car en général, le nombre total de modalités conservées est nettement inférieur. J'ai donc décidé, pour gérer cette différence de limitation de manière suffisamment simple, d'écrire deux programmes :

- le premier comportera la saisie, la validation et le nettoyage des données, ainsi que la sélection des variables discriminantes et il permettra de traiter un grand nombre de modalités.
- le deuxième, écrit pour un nombre de modalités plus "restreint", sera uniquement consacré à l'analyse factorielle et au classement des individus. De plus, cette découpe permet d'effectuer plusieurs analyses sans devoir chaque fois recommencer le travail fastidieux du nettoyage des données et de la sélection des variables.

CHAPITRE VI :

- ENTREES - SORTIES -

VI.1 Généralités.

La saisie et la validation des données sont du ressort du premier programme. Les données sont divisées en quatre parties dont une qui est facultative. Les trois premières, qui se trouvent toujours sur fichier sont :

- 1) les poids des individus
- 2) les codages condensés
- 3) les libellés des variables et des modalités (facultatif).

Pour les autres données, j'ai laissé le choix à l'utilisateur entre, soit les introduire par l'intermédiaire d'un fichier, soit les entrer directement au terminal. Remarquons que, si la lecture se fait sur fichier, aucune correction n'est possible en cas d'erreur, tandis que pour la saisie à partir d'un terminal, on peut redemander une donnée pour laquelle on a découvert une erreur, et en fait, on ne passera à la lecture de la donnée suivante que si les précédentes sont correctes. Chaque détection d'erreur sera accompagnée de l'émission d'un message au terminal, message différent selon qu'il s'agira d'un problème lors de la lecture d'une donnée ou lors de la validation de celle-ci; il y aura un message par donnée pour pouvoir localiser l'erreur en cas de saisie par fichier.

Les outputs du premier programme sont également divisés en deux catégories : la première consiste en un ensemble de fichiers contenant les informations nécessaires à l'exécution du second programme, et dont un, le fichier "principal", contient entre autres, les noms de tous les autres; la deuxième consiste en un ensemble de résultats inemployés par la suite et résultant du nettoyage et de la sélection. La sortie de cette dernière partie peut s'effectuer soit sur imprimante, soit au terminal, soit dans un fichier.

Pour exécuter le second programme, l'utilisateur ne devra introduire que le nom du fichier "principal" résultant de l'exécution du premier. La sortie des résultats s'effectuant de la même manière que pour le premier programme.

VI.2 Données générales nécessaires à l'exécution du premier programme.

=====

Une des premières indications à fournir est :

i.1) le nombre total de variables explicatives à considérer (NV).

NV devra être supérieur ou égal à un pour que le problème ait un sens et il devra être inférieur ou égal à une certaine borne pour des raisons pratiques. J'ai fixé cette borne à 100 car cela permet de traiter à peu près toutes les applications et cela ne pose pas de problème de place mémoire compte tenu de la limitation à 9 modalités au maximum par variable (voir page V.3).

Ce programme devant permettre de faire des expériences comme, par exemple, exécuter l'analyse sur différents groupes de variables, ces variables ne sont pas forcément les NV premières dans le tableau des codages condensés. On donnera donc :

i.2) le mode de répartition des variables dans ce tableau (MODVAR).

MODVAR n'a en fait que deux valeurs acceptables : MODVAR1 si les variables ne sont pas les NV premières et MODVAR2 sinon. Si MODVAR prend la valeur MODVAR1, il faudra évidemment donner en plus :

i.3) les numéros des variables à considérer (NUMVAR(1), ..., NUMVAR(NV)).

On pourrait donner ces numéros dans n'importe quel ordre mais, pour raison de clarté, j'ai imposé qu'ils soient donnés par ordre croissant et que NUMVAR(NV) n'excède pas la borne supérieure sur le nombre de variables,

ce qui limite le nombre de colonnes du tableau des codages condensés à cette borne.

Pour pouvoir lire et valider le contenu du fichier des poids des individus, il faut :

i.4) le nombre total d'individus (NI).

NI doit être supérieur ou égal à 2 (pour que 2 groupes au moins soient représentés) et inférieur ou égal à une certaine borne NIMAX. J'ai fixé NIMAX à 4000.

i.5) le nom du fichier des poids des individus.

Concernant les libellés des variables et des modalités, plusieurs cas sont à envisager :

- soit l'utilisateur a mis ceux-ci dans un fichier, par exemple sous la forme

- - -

libellé de la variable n° i, nombre de modalités

libellés des modalités de la variable n° i

- - -

- soit il ne l'a pas fait et, dans ce cas, les variables et les modalités sont repérées par des numéros.

Nous aurons donc :

i.6) un indicateur de l'existence d'un fichier de libellés

et, si cet indicateur prend la valeur correspondant à l'existence d'un tel fichier,

i.7) le nom du fichier des libellés.

Afin d'éviter de perdre du temps dans le cas où les variables ont exactement les mêmes libellés de modalités, ce fichier pourra avoir deux structures différentes. La première, qui correspondra au cas où les libellés des

modalités sont identiques pour les variables, sera :

ligne 1 : nombre de modalités (le même pour chaque variable)

ligne 2 : libellés des modalités

pour i allant de 1 à NV,

ligne i+2 : libellé de la variable n° i

tandis que pour la deuxième, nous aurons :

pour i allant de 1 à NV :

ligne 2.i-1 : libellé de la variable n° i et nombre de modalités pour celle-ci

ligne 2.i : libellés des modalités de la variable n° i.

La saisie du contenu de ce fichier nécessite donc de connaître la valeur d'un

i.8) indicateur de l'égalité des libellés des modalités des variables.

S'il n'emploie pas de fichier de libellés, l'utilisateur doit introduire le nombre de modalités de chaque variable. Pour ne pas répéter NV fois la même chose en cas d'égalité de ceux-ci, il donnera :

i.9) l'indicateur de l'égalité du nombre de modalités de chaque variable.

S'il n'y a pas égalité, il lui faudra introduire

i.10) la liste des nombres de modalités

et, dans le cas contraire,

i.11) le nombre de modalités.

Une des principales données d'un programme d'analyse, est ce que l'on appelle le "tableau des codages condensés". C'est un tableau A tel que $A(i,j)=k$ si et seulement si le $i^{\text{ième}}$ individu prend la $k^{\text{ième}}$ modalité de la $j^{\text{ième}}$ variable. Un choix est à faire concernant la variable Y qui, rappelons le, détermine les groupes d'individus :

- ou bien on considère que c'est une variable comme une autre et, de ce fait,

on l'inclu dans le tableau A (en première ou en dernière colonne par exemple)

- ou bien, on estime qu'elle joue un rôle différent et, dans ce cas, on peut la mettre à part en adoptant par exemple un mode de représentation sous forme d'un tableau à une dimension dont la $i^{\text{ième}}$ composante représente le nombre d'individus appartenant au $i^{\text{ième}}$ groupe (ceci nécessite que les observations relatives à des individus d'un même groupe se suivent dans le tableau des codages condensés). Malgré cette restriction, j'ai opté pour la deuxième solution car elle minimise le nombre de modifications à apporter aux données lors d'un changement dans les groupes d'individus (par exemple en cas d'agrégation de deux classes). Les trois données à introduire sont donc :

i.12) le nom du fichier des codages condensés

i.13) le nombre de groupes d'individus

i.14) la liste des nombres d'individus dans chaque groupe.

Ce programme permettant de classer des individus supplémentaires, on indiquera

i.4.bis) le nombre d'individus supplémentaires à classer.

Les codages condensés des modalités prises par ces individus pour chaque variable devant subir le même "nettoyage" que ceux correspondant aux individus de base, j'ai choisi de les mettre dans le même fichier à la suite de ceux-ci.

Une grande liberté étant laissée à l'utilisateur quant à la méthode qu'il peut employer pour boucher les trous dans le tableau des codages condensés et quant au moment auquel il peut effectuer son choix, il lui faudra donner

i.15) l'indicateur de quand il choisit la méthode.

Si le choix s'effectue avant le nettoyage, il introduira en plus

i.16) l'indicateur du choix de la méthode.

La troisième méthode nécessite l'introduction des modalités groupes pour les modalités de chaque variable. Si les regroupements sont identiques (ceci n'est possible que si les variables ont même nombre de modalités) il est inutile de répéter NV fois la même chose. C'est pourquoi j'ai prévu deux modes d'introduction suivant la valeur que l'utilisateur donnera à

i.17) l'indicateur de l'égalité des regroupements.

S'il y a égalité, il suffira d'introduire

i.18) le vecteur des modalités groupes des modalités

et, dans le cas contraire, il faudra introduire ce vecteur pour chaque variable :

i.19) pour i allant de 1 à NV :

le vecteur des modalités groupes des modalités de la variable n° i.

Pour pouvoir effectuer le nettoyage des données, il ne manque plus que :

i.20) le poids maximum autorisé pour la modalité "non-répondu" d'une variable

i.21) le poids minimum pour qu'une modalité soit conservée.

Le choix de la méthode employée pour le test d'arrêt lors de la sélection étant libre, l'utilisateur doit introduire deux des trois choses suivantes :

i.22) l'indicateur du choix de la méthode d'arrêt.

Si c'est la méthode par nombre fixé de variables à sélectionner,

i.23) le nombre de variables à sélectionner

et sinon

i.24) la quantité fixant la croissance minimale pour le coefficient de Tschuprow multiple.

La dernière donnée nécessaire à la sélection proprement dite est :

i.25) le seuil à partir duquel on considère que deux variables sont redondantes.

Si on ajoute à cela :

i.26) l'inertie minimale pour qu'un facteur z soit conservé

i.27) le pouvoir discriminant cumulé à atteindre pour les facteurs z ,

le reste des données à introduire se rapporte au "quoi" et au "où" des outputs.

Nous avons tout d'abord une série de noms de fichiers.

i.28) le nom du fichier des codages condensés après sélection.

Ce fichier sert au reclassement des individus de départ et au classement des éventuels individus supplémentaires.

Si un fichier de libellés est utilisé, nous aurons également

i.29) le nom du fichier des libellés après sélection.

L'évaluation du tableau de Burt avant la sélection nécessitant un fichier, nous aurons :

i.30) le nom du fichier devant contenir le tableau de Burt croisant les modalités des variables explicatives avant la sélection.

L'analyse discriminante nécessitant ce tableau après sélection ainsi que celui

croisant les modalités des variables explicatives avec celles de Y, suivent donc :

- i.31) le nom du fichier devant contenir le tableau de Burt croisant les modalités des variables explicatives entre elles après la sélection
- i.32) le nom du fichier devant contenir le tableau de Burt croisant les modalités des variables explicatives avec celles de Y après la sélection.

Les données pour le second programme étant toujours lues à partir d'un fichier créé par le premier, il faudra

- i.33) le nom du fichier des données générales pour le second programme.

Comme je l'ai déjà mentionné, l'utilisateur a le choix entre trois modes de sortie pour les résultats intermédiaires :

```

mode 1 --- terminal
mode 2 --- imprimante
mode 3 --- fichier

```

Il devra donc donner

- i.34) l'indicateur du mode de sortie des résultats
- et, si c'est le mode 3 qui est choisi
- i.35) le nom du fichier des résultats intermédiaires.

Après ce groupe de données, viennent un certain nombre d'informations complémentaires concernant les résultats que l'utilisateur désire voir édités. Si la liste des modalités de poids trop faible est donnée d'office pour pouvoir se rendre compte de ce qui a été éliminé, il n'en est pas de même pour la liste des poids des modalités après nettoyage qui n'a qu'un rôle secondaire. L'utilisateur introduira donc

- i.36) l'indicateur d'édition de la liste des poids des modalités après nettoyage.

Il en est de même pour un certain nombre de choses qui ne sont pas vraiment indispensables comme, par exemple, le tableau des coefficients, la valeur du coefficient de Tschuprow multiple, etc Il faudra donc préciser si on désire ou on ne désire pas :

- i.37) le tableau des coefficients de Tschuprow du premier ordre entre les variables
- i.38) le vecteur des coefficients de Tschuprow du premier ordre avec Y
- i.39) la valeur du coefficient de Tschuprow maximum avec Y à chaque étape
- i.40) la valeur du coefficient de Tschuprow multiple et de sa croissance à chaque étape.

Pour ne pas avoir un volume de résultats trop important, on peut également demander les coefficients de Tschuprow d'ordre k entre les variables ou entre les variables et Y, non pas à chaque étape, mais toutes les F étapes après la première. On introduira donc

- i.41) la fréquence avec laquelle on désire le tableau des coefficients de Tschuprow entre les variables
- i.42) la fréquence avec laquelle on désire le tableau des coefficients de Tschuprow entre les variables et Y.

VI.3 Résultats à attendre de l'exécution du premier programme.

=====

VI.3.1 : Le fichier contenant le tableau de Burt croisant les modalités des variables explicatives entre elles après nettoyage et avant sélection.

Ce fichier, utilisé pour le calcul des coefficients de Tschuprow du premier ordre n'est plus employé par la suite lors de l'analyse discriminante. Mais, comme il peut être utile pour d'autres types

d'analyses, j'ai décidé de ne pas le supprimer après la phase initiale de la sélection. Le tableau de Burt est rangé tout entier, ligne par ligne, dans ce fichier.

VI.3.2 : Le fichier contenant le tableau de Burt croisant les modalités des variables explicatives après la sélection.

Ce tableau sert lors de l'analyse des correspondances effectuée à la première étape de l'analyse discriminante. Plutôt que de le recalculer à partir des codages condensés lors de l'exécution du deuxième programme, j'ai préféré le mettre à jour en fonction des variables sélectionnées et le stocker sur fichier.

VI.3.3 : Le fichier contenant le tableau de Burt croisant les modalités des variables explicatives avec celles de Y après la sélection. Pour raison de clarté, je l'ai stocké non sous la forme $X^t D_p Y$, mais sous sa forme transposée. Ce tableau est utilisé lors du calcul des pouvoirs discriminants des facteurs z et lors de la formation de la matrice B_z . Il sert également au classement des individus.

VI.3.4 : Le fichier contenant le tableau des codages condensés après la sélection.

Ce tableau a la même forme que le tableau de départ excepté qu'il contient un nombre de colonnes exactement égal au nombre de variables qui ont été sélectionnées et un nombre de lignes égal au nombre d'individus conservés (individus de départ + individus supplémentaires). Il est employé pour le classement de tous ces individus.

VI.3.5 : Le fichier de données pour le second programme.

Il doit contenir :

- le nombre de variables

- les positions des modalités des variables
 - le nombre d'individus de base
 - le nombre d'individus supplémentaires
 - les poids des individus
 - le nombre de groupe d'individus
 - le nombre d'individus dans chaque groupe
 - les noms des fichiers VI.3.2 à VI.3.4
 - l'indicateur du mode de sortie des résultats
- et, si c'est le mode fichier, le nom du fichier des résultats intermédiaires du premier programme (on y ajoutera ceux du second programme)
- l'inertie minimale pour conserver un facteur z
 - le pouvoir discriminant cumulé à atteindre par les facteurs z.

VI.3.6 : Le fichier des résultats intermédiaires.

Outre ce qui a été demandé via les données générales, on y trouvera :

- un rappel de certaines données :
 - le nombre de variables au départ
 - les numéros des variables
 - le nombre d'individus au départ
 - le nombre d'individus supplémentaires
 - les poids des individus
 - éventuellement le nom du fichier des libellés avant nettoyage
 - les nombres de modalités pour chaque variable
 - le nom du fichier des codages condensés au départ
 - le nombre de groupes d'individus
 - le nombre d'individus dans chaque groupe
 - le poids maximum autorisé pour la modalité "non-répondu"
 - ...

- des résultats du nettoyage :

- la liste des modalités de poids trop faible
- les éventuels regroupements de modalités
- la liste des variables éliminées pour poids trop important de la modalité "non-répondu"
- la liste des variables éliminées pour nombre de modalités trop faible
- la liste des individus supprimés

- des résultats de la sélection :

- la liste des variables éliminées pour redondance à chaque étape de la sélection :
- le numéro de l'étape
- l'indice de la variable sélectionnée
- s'il y a arrêt, la cause de cet arrêt
- la liste des variables éliminées pour coefficient de Tschuprow d'ordre k proche de 1 avec la variable sélectionnée.

VI.4 Données et résultats du deuxième programme.

=====

Comme je l'ai déjà mentionné, il suffira de lire le nom du fichier de données résultant de l'exécution du premier programme pour avoir tout ce qui est nécessaire au deuxième. Les résultats à attendre de celui-ci sont :

- les codages descriptifs et les inerties car ils ont une interprétation intéressante
- les pouvoirs discriminants des différents facteurs
- les numéros des facteurs \underline{z} conservés avec le pouvoir discriminant cumulé atteint
- les codages discriminants pour les facteurs \underline{z} ainsi que les pouvoirs discriminants associés

- les codages discriminants recherchés
- pour chaque groupe de départ, le résultat du reclassement de ses individus
- le résultat du classement des individus supplémentaires.

CHAPITRE VII :

- DESCRIPTION DES PROGRAMMES -

VII.1 Description du premier programme.

=====

Rappelons qu'il est chargé d'effectuer :

- la saisie et la validation des données
- le nettoyage des données
- la sélection des variables explicatives

suivant les méthodes et les remarques exposées dans les chapitres précédents.

Le programme principal s'appelle PRINCl et se trouve dans le fichier

PRINCl.FOR. Les procédures et fonctions nécessaires à son exécution sont :

Procédures :

SAISIE (dans SAISIE.FOR)

NETTOY (dans NETTOY.FOR)

SELECT (dans SELECT.FOR)

MSGT
MSGERR (dans MSG.FOR)

SAVA1
SAVA2
SAVA3
SAVA4
SAVA5
SAVA6
SAVA7
SAVA9
SAVA10
SAVA11
SAVA12
SAVA13
SAVA14 (dans SAVA.FOR)

LEC1
LEC2
LEC3
LEC4
LEC5
LEC6
LEC7
LEC8 (dans LECTUR.FOR)

RINTAB (dans SELECT.FOR)

Fonctions :

ROUTAB (dans SELECT.FOR)

ZVLD1

ZVLD2

ZVLD3

ZVLD4

ZVLD5

ZVLD6

ZVLD7

ZVLD8

ZVLD9

ZVLD10

ZVLD11

ZVLD12

ZVLD13

ZVLD14

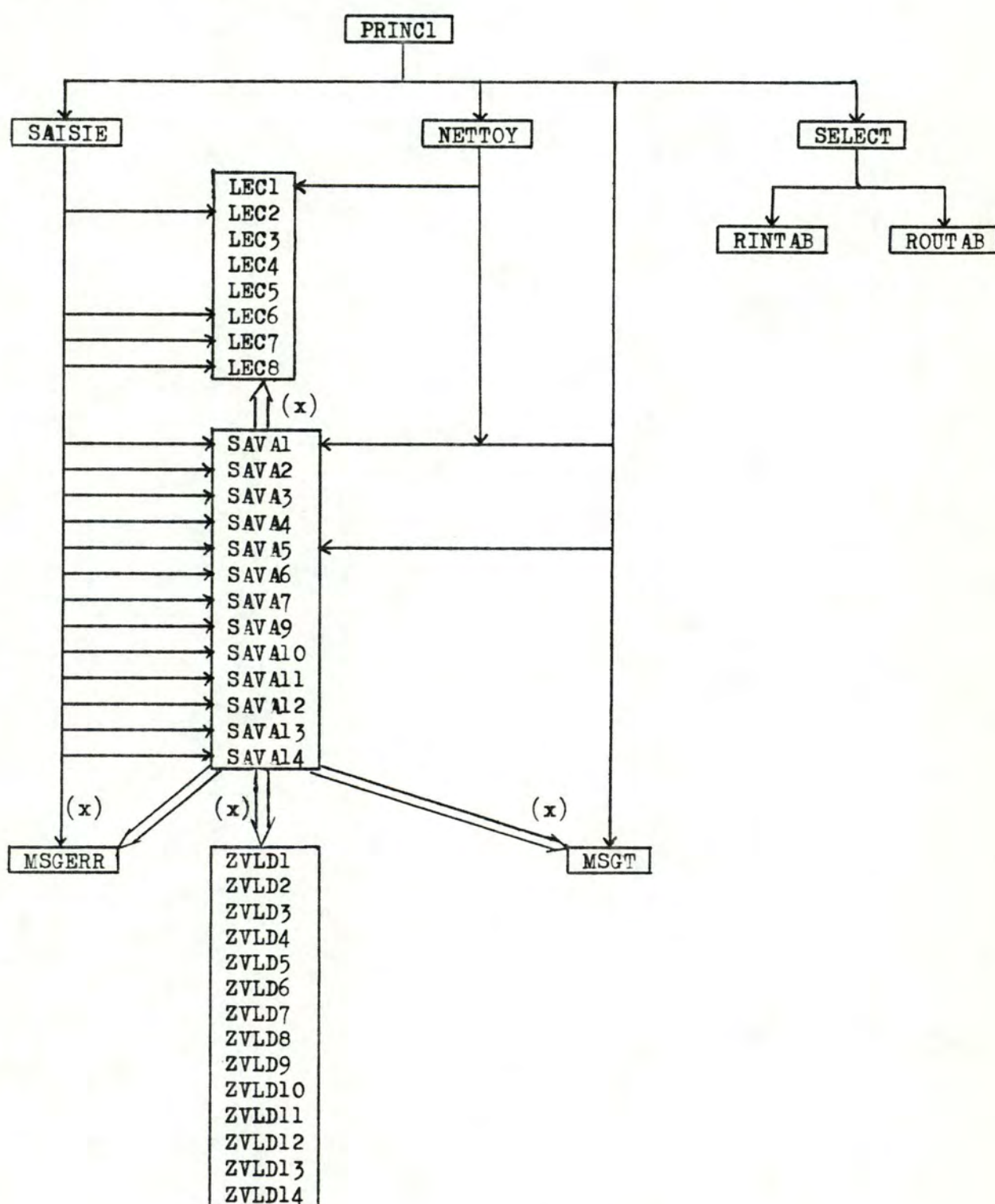
(dans VALIDA.FOR)

Description sommaire.

En gros, on peut dire que PRINCl appelle successivement SAISIE (pour la saisie et la validation des données), NETTOY (pour le nettoyage des données) et SELECT (pour la sélection des variables explicatives). Les procédures LECi servent à la lecture tandis que les fonctions ZVLDi retournent la valeur .TRUE. ou .FALSE. suivant qu'une donnée ne vérifie pas ou bien vérifie une certaine contrainte. Les procédures SAVAi sont chargées de la saisie et de la validation à un niveau supérieur (saisie sur fichier ou saisie au terminal avec redemande en cas de détection d'erreur). Dans MSG.FOR on trouve des routines d'affichage de messages d'introduction (MSGT) et de messages d'erreur (MSGERR). ROUTAB et RINTAB s'occupent de conversions d'indices suivant que l'on stocke une matrice symétrique dans un vecteur ou non.

On peut trouver le schéma des appels à la page VII.3 et les descriptions de PRINCl, des procédures et des fonctions aux pages suivantes.

Schéma des appels :



(x): voir SAVA.FOR pour le détail des appels.

VII.1.1 Description de PRINCl (PRINCl.FOR).

Identificateurs employés pour les variables, tableaux et constantes.

Constantes communes à PRINCl, SAISIE, SELECT et à NETTOY (jusqu'à IPMN2).

COMMON/COM1/

- ILIB1 : Valeur que ILIB doit prendre quand il n'y a pas utilisation d'un fichier de libellés.
- ILIB2 : Idem dans le cas contraire.
- IMAR1 : Valeur que IMAR doit prendre quand on veut fixer le nombre de variables à sélectionner.
- IMAR2 : Idem quand on choisit l'autre méthode d'arrêt.
- IQDBT1 : Valeur que doit prendre IQDBT quand on ne choisit pas la méthode de nettoyage avant le début de celui-ci.
- IQDBT2 : Idem dans le cas contraire.
- ICHTR1 : Valeur que doit prendre ICHTR quand on choisit la méthode par suppression des individus.
- ICHTR2 : Idem quand on choisit celle par ventilation.
- ICHTR3 : Idem quand on choisit celle par regroupements.
- IPMN1 : Valeur que doit prendre IPMN si on ne désire pas la liste des poids des modalités après nettoyage.
- IPMN2 : idem dans le cas contraire.
- ICT11 : Valeur que doit prendre ICT1 si on ne veut pas avoir le tableau des coefficients de Tschuprow du premier ordre entre les variables.
- ICT12 : Idem dans le cas contraire.
- ICTY11 : Idem ICT11 mais pour les coefficients avec Y.
- ICTY12 : Idem ICT12 mais pour les coefficients avec Y.
- ICTMX1 : Valeur que doit prendre ICTMX si on ne veut pas la valeur du coefficient de Tschuprow maximum avec Y à chaque étape.
- ICTMX2 : Idem si on désire cette valeur.
- ICTMP1 : Valeur que doit prendre ICTMP si on ne désire pas la valeur du coefficient de Tschuprow multiple et sa croissance à chaque étape.
- ICTMP2 : Idem dans le cas contraire.

- NOUT1 : Valeur que doit prendre NOUT si on veut les résultats intermédiaires sur terminal.
- NOUT2 : Idem si on les veut sur imprimante.
- NOUT3 : Idem si on les veut sur fichier.

Variables communes à PRINCl, SAISIE et SELECT.

COMMON/COM2/

- ICT1 : Indicateur d'édition du tableau des coefficients de Tschuprow du premier ordre entre les variables explicatives.
- ICTY1 : Idem pour les coefficients entre les variables explicatives et Y.
- ICTMX : Indicateur d'édition du coefficient de Tschuprow maximum à chaque étape.
- ICTMP : Idem pour le coefficient de Tschuprow multiple et sa croissance.
- IMAR : Indicateur du choix de la méthode d'arrêt.
- ICTF : Période avec laquelle on veut le tableau des coefficients de Tschuprow d'ordre supérieur entre les variables explicatives.
- ICTYF : Idem pour le tableau des coefficients entre les variables explicatives et Y.
- SR : Seuil à partir duquel on considère que deux variables sont redondantes.
- QTM : Valeur fixant la croissance minimale pour le coefficient de Tschuprow multiple.
- NVS : Nombre de variables à sélectionner.
- VLPMIN : Inertie minimale pour qu'un facteur z soit conservé.
- PDISCU : Pouvoir discriminant cumulé à atteindre par les facteurs z.

Variables de type CHARACTER communes à PRINCl, SAISIE et SELECT.

COMMON/COM3/

- FCODN : (C) Le nom du fichier des codages condensés après sélection.
- FLIBN : (C) Le nom du fichier des libellés après sélection.
- FBURT : (C) Le nom du fichier contenant le tableau de Burt croisant les modalités des variables explicatives entre elles après la sélection.
- FDON2 : (C) Le nom du fichier de données pour le second programme.
- FBURTA : (C) Idem FBURT mais avant la sélection (après le nettoyage).

FBURTY : (C) Le nom du fichier contenant le tableau de Burt croisant les modalités des variables explicatives avec celles de Y après la sélection.

Variables communes à PRINCl, NETTOY et SELECT.

COMMON/COM6/

NUMRES : Numéro d'unité pour la sortie des résultats intermédiaires.

NTOTM : Nombre total de modalités.

Variables de type CHARACTER communes à PRINCl, NETTOY et SELECT.

COMMON/COM7/

FCOD1 : (C) Le nom du fichier contenant le tableau des codages condensés après le nettoyage et avant la sélection.

Variables communes à PRINCl, SAISIE et NETTOY.

COMMON/COM4/

PNRV : Poids maximum autorisé pour la modalité non-répondu d'une variable.

PMMIN : Poids minimum pour qu'une modalité soit conservée.

IQDBT : Indicateur de quand on choisit la méthode pour boucher les trous dans le tableau des codages condensés.

ICHTR : Indicateur du choix de la méthode pour boucher les trous dans le tableau des codages condensés.

IPMN : Indicateur d'édition des poids des modalités après nettoyage.

MODGR : Vecteur dont la $i^{i\text{ème}}$ composante est la modalité groupe de la $i^{i\text{ème}}$ modalité.

NUMTER : Numéro d'unité pour TTY.

Variables de type CHARACTER communes à PRINCl, SAISIE et NETTOY.

COMMON/COM8/

FCOD : (C) Nom du fichier des codages condensés avant nettoyage.

FLIB : (C) Nom du fichier des libellés avant nettoyage.

Variables communes à PRINCl, SAISIE, NETTOY et SELECT.

COMMON/COM5/

NV : Nombre de variables.

NUMVR : Numéros de variables.
 NI : Nombre d'individus.
 NICLA : Nombre d'individus supplémentaires.
 PI : Vecteur des poids des individus.
 ILIB : Indicateur de l'existence d'un fichier de libellés.
 MPOS : Vecteur des positions des modalités des différentes variables.
 MPOS(I)=K \Leftrightarrow La première modalité de la variable n° I se trouve
 en position K+1.
 MPOS(NV+1) = NTOTM.
 NGR : Nombre de groupes d'individus.
 NVIGR : Vecteur des nombres d'individus dans chaque groupe.

Variables de type CHARACTER communes à PRINCl, SAISIE, NETTOY et SELECT.

COMMON/COM9/

LIBVAR : (C) Vecteur des libellés des variables.
 LIBMOD : (C) Vecteur des libellés des modalités.

Autres identificateurs de variables et de constantes.

FRES : (C) Nom du fichier des résultats.
 FDON : (C) Nom du fichier des données générales.
 NCXIN : Indicateur du choix du mode d'entrée.
 NCXIN1 : Valeur que doit prendre NCXIN si un fichier est employé pour
 introduire les données.
 NCXIN2 : Idem mais si on introduit les données directement au terminal.
 NUMFD : Numéro d'unité pour le fichier de données.
 NUMFR : " " " " " des résultats.
 NUMIN : " " " l'introduction des données.
 NUMLPT : " " pour l'imprimante.
 NOUT : Indicateur du choix du mode de sortie.
 ZP : Indicateur d'erreur pour la saisie de NCXIN et de FDON.
 ZTOTAL : Indicateur principal d'erreur.

Concernant les types.

Que ce soit pour PRINCl ou pour n'importe quelle procédure ou fonction, les trois règles suivantes sont d'application.

Règle 1 : Les identificateurs commençant par la lettre Z sont des identificateurs de variables, constantes, tableaux ou fonctions de type LOGICAL.

Règle 2 : Les identificateurs commençant par une des lettres I,J,K,L,M ou N se rapportent à des ENTIERS.

Règle 3 : Les identificateurs commençant par une des lettres A,...,H et O,...,Y concernent des DOUBLE PRECISION.

Les seules exceptions à ces règles sont les variables de type CHARACTER (C).

Sous-programmes appelés par PRINCl.

MSGT

SAVA1

SAVA2

SAISIE

NETTOY

SELECT

Concernant les dimensions des tableaux.

Si vous désirez changer certaines bornes, il faudra toujours veiller à ce que les tableaux aient au moins les dimensions suivantes :

MODGR (NTOTM)
 LIBMOD (NTOTM)
 NUMVR (NV)
 LIBVAR (NV)
 PI (NI)
 MPOS (NV+1)
 NVIGR (NGR)

Concernant l'algorithme.

- - - - -

Il est vraiment très simple et comprend les étapes suivantes :

1) Affichage du message d'introduction

Saisie de l'indicateur du choix du mode d'entrée et, si nécessaire
saisie du nom du fichier de données.

2) Appel à SAISIE pour la saisie et la validation des données.

S'il y a eu un problème lors de la saisie ou de la validation, affichage
d'un message d'erreur et arrêt du programme.

3) Détermination du mode de sortie des résultats (Il s'agit de donner
la valeur adéquate à NUMRES).

4) Appel à NETTOY pour effectuer le nettoyage des données.

S'il y a eu un problème lors du nettoyage ou si le nombre de groupes
d'individus restants est inférieur à 2 ou s'il ne subsiste plus de
variable, affichage du message approprié et arrêt du programme.

5) Appel à SELECT pour effectuer la sélection des variables discriminantes.

VII.1.2 Description de SAISIE (SAISIE.FOR).

Arguments.

ZTOTAL : Indicateur principal d'erreur.

NOUT : Indicateur du choix du mode de sortie.

FRES : (C) Nom du fichier des résultats intermédiaires.

NUMIN : Numéro d'unité pour la lecture des données.

Identificateurs employés pour les variables, tableaux et constantes.

Identificateurs se trouvant dans les listes de COMMON.

voir PRINCl.

Identificateurs de constantes locales à SAISIE.

NUMFP : Numéro d'unité pour le fichier des poids des individus.

NUMFL : " " " " " " libellés.

NUMCC : " " " " " " codages condensés.

NVMIN : Minimum pour NV.

NVMAX : Maximum pour NV.

NVSMIN : Minimum pour NVS.

NIMIN : Minimum pour NI.

NIMAX : Maximum pour NI.

NICLMI : Minimum pour NICLA.

MODMIN : Nombre minimum de modalités pour une variable.

MODMAX : Nombre maximum de modalités pour une variable.

NGRMIN : Nombre minimum de groupes d'individus.

NGRMAX : Nombre maximum de groupes d'individus.

PNRMIN : Minimum pour PNRV.

PNRMAX : Maximum pour PNRV.

PMMINF : Infimum pour PMMIN.

PMMSUP : Suprémum pour PMMIN.
 EPSPMM : Précision sur PMMIN. Il faut que $PMMINF + EPSPMM$ soit inférieur ou égal à PMMIN et que $PMMSUP - EPSPMM$ soit supérieur ou égal à PMMIN.
 ITPGR1 : Valeur que doit prendre ITPGR si les modalités groupées des modalités des variables ne sont pas les mêmes pour toutes les variables.
 ITPGR2 : Idem mais dans le cas contraire.
 MVAR1 : Valeur que doit prendre MODVAR si les variables ne se suivent pas dans le fichier des codages condensés.
 MVAR2 : Idem dans le cas contraire.
 NELIB1 : Valeur que doit prendre NELIB si les libellés des modalités ne sont pas les mêmes pour toutes les variables.
 NELIB2 : Idem mais dans le cas contraire.
 NEMOD1 : Valeur que doit prendre NEMOD si le nombre de modalités n'est pas le même pour chaque variable.
 NEMOD2 : Idem mais dans le cas contraire.
 PIINF : Infimum pour le poids d'un individu.
 PISUP : Suprémum pour le poids d'un individu.
 EPSPI : Précision sur le poids d'un individu.
 SOMPI : Valeur que doit donner la somme des poids des individus.
 EPSSPI : Précision sur SOMPI.
 QTMINF : Infimum pour QTM.
 QTMSUP : Suprémum pour QTM.
 EPSQTM : Précision sur QTM.
 SRINF : Infimum pour SR.
 SRSUP : Suprémum pour SR.
 EPSSR : Précision sur SR.
 ICTFMM : Minimum pour ICTF.
 ICTYFM : Minimum pour ICTYF.
 VLPINF : Infimum pour VLPMIN.
 VLPSUP : Suprémum pour VLPMIN.
 EPSVLP : Précision sur VLPMIN.

PDCINF : Infimum pour PDISCU.

EPSPDC : Précision sur PDISCU.

Identificateurs de variables simples.

ITPGR : Indicateur de l'égalité des modalités groupes des modalités pour toutes les variables.

MODVAR : Indicateur du mode de répartition des variables dans le fichier des codages condensés.

NBMOD : Variable représentant un nombre de modalités.

NELIB : Indicateur de l'égalité des libellés des modalités pour toutes les variables.

NEMOD : Indicateur de l'égalité du nombre de modalités pour toutes les variables.

FPDS : (C) Nom du fichier des poids des individus.

ZNV, ZMVAR, ZNUMVR, ZNI, ZNICLA, ZFPDS, ZILIB, ZFLIB, ZNELIB, ZNEMOD, ZNBMOD, ZNVMOD, ZFCOD, ZGR, ZNVIGR, ZIMAR, ZQTM, ZNVS, ZPNRV, ZPMMIN, ZIQDBT, ZICHTR, ZITPGR, ZNREGR, ZSR, ZFCODN, ZFLIBN, ZBURTA, ZFBURT, ZBURTY, ZFDON2, ZVLP, ZPDC, ZNOUT, ZFRES, ZIPMN, ZICT1, ZICTF, ZICTY1, ZICTYF, ZICTMX, ZICTMP : Indicateurs d'erreurs

- soit de problème lors de la lecture
- soit de non validation ou d'erreur détectée lors de la validation
- soit de non affectation de certains vecteurs suite à des problèmes

respectivement pour les variables NV, MODVAR, NUMVR, NI, NICLA, FPDS, ILIB, FLIB, NELIB, NEMOD, NBMOD, NVMOD, FCOD, NGR, NVIGR, IMAR, QTM, NVS, PNRV, PMMIN, IQDBT, ICHTR, ITPGR, NREGR, SR, FCODN, FLIBN, FBURTA, FBURT, FBURTY, FDON2, VLP, PDISCU, NOUT, FRES, IPMN, ICT1, ICTF, ICTY1, ICTYF, ICTMX, ICTMP.

ZPI : Indicateur d'erreur dans le fichier des poids des individus.

ZCLIB : Indicateur d'erreur dans le fichier des libellés.

ZCODC : Indicateur d'erreur dans le fichier des codages condensés.

Identificateurs de tableaux.

LAUMOD : Vecteur contenant les libellés des modalités d'une variable. (C)

NCODI : Vecteur contenant les codages condensés correspondant à un individu.

NREGR : Vecteur contenant les modalités groupes des modalités d'une variable.

NVMOD : Vecteur des nombres de modalités pour les variables.

Concernant les dimensionnements.

```
LAUMOD (MODMAX)
NREGR (MODMAX)
NVMOD (NV)
NCODI (NUMVR(NV)).
```

Sous-programmes appelés par SAISIE.

```
SAVA1
SAVA2
SAVA3
SAVA4
SAVA5
SAVA6
SAVA7
SAVA9
SAVA10
SAVA11
SAVA12
SAVA13
SAVA14
```

```
MSGERR
```

```
LEC2
LEC6
LEC7
LEC8
```

Effet de SAISIE.

Action sur les arguments :

NUMIN est inchangé.

Si une erreur a été détectée dans les données, ZTOTAL aura la valeur .TRUE.

et sinon, il aura la valeur .FALSE. et dans ce cas :

NOUT aura une des trois valeurs NOUT1, NOUT2 ou NOUT3.

Si NOUT=NOUT3, FRES contiendra le nom du fichier des résultats intermédiaires.

Action sur les variables communes :

Si ZTOTAL a la valeur .FALSE., les variables déclarées dans les communs COM2, COM3, COM4, COM5, COM8, et COM9 auront des valeurs conformes aux contraintes sur les données (cfr. Annexe C) sauf peut-être pour les noms de fichiers.

VII.1.3 Description de NETTOY (NETTOY.FOR).

Argument.

ZTOTAL : Indicateur principal d'erreur.

Identificateurs employés pour les variables, tableaux et constantes.

Identificateurs se trouvant dans les listes de COMMON.

Voir PRINCl.

Identificateurs de constantes locales à NETTOY.

NUMCC : Numéro d'unité pour les accès au fichier FCOD.

NUMCC1 : Numéro d'unité pour les accès au fichier FCOD1.

NUMCC2 : Numéro d'unité pour les accès au fichier FCOD2.

Identificateurs de variables simples.

ICHOIX : Indicateur du choix de la méthode à employer pour la modalité traitée.

IREV : Indicateur d'affichage à l'écran des poids des modalités non déjà éliminées et des nombres d'individus concernés par celles-ci pour la variable dont on traite une modalité de poids trop faible.

NAU : Indice d'une modalité d'une variable (autre que celle que l'on traite) prise par l'individu qui doit être supprimé.

NBT : Nombre d'individus trouvés qui prennent la modalité de poids trop faible que l'on traite

NICAU : Nombre d'individus concernés par la modalité de poids trop faible que l'on traite.

NIG : Nombre d'individus à conserver.

NJUM : Numéro d'une modalité non éliminée à laquelle jumeler la modalité de poids trop faible que l'on traite.

NMABAN : Nombre de modalités abandonnées.

NMGT : Nombre de modalités à garder au total.

NMGV : Nombre de modalités à garder pour la variable que l'on est en train de traiter.

NVG : Nombre de variables à conserver.

- PAJOUT : Poids à rajouter à chaque individu restant après la suppression des individus se rapportant à une modalité de poids trop faible.
- PAUX : Poids d'une modalité autour de laquelle on effectue un regroupement.
- PPLUS : Somme des poids des individus concernés par la modalité à éliminer.
- FCOD2 : (C) Nom du fichier intermédiaire des codages condensés, nécessaire lorsqu'on emploie la méthode de nettoyage par ventilation.
- ZFIN : Indicateur de fin de recherche lorsqu'on emploie la méthode par élimination des individus ou lorsque le nettoyage se fait à la carte.
- ZIMP : Indicateur d'impression de titre.
- ZL : Indicateur d'erreur pour le numéro de la modalité à laquelle on veut jumeler celle de poids insuffisant que l'on traite.
- ZP : Indicateur d'erreur pour IREV.
- ZPGM : Indicateur de l'existence d'une modalité de poids suffisant pour une variable lorsque l'on emploie la méthode par ventilation.
- ZPGR : Indicateur d'insuffisance de poids après un regroupement.
- ZTR : Indicateur de réussite pour la recherche d'une modalité de poids suffisant lors de la ventilation.

Identificateurs de tableaux.

- MPMG : Vecteur indicateur des modalités à conserver.
 Au départ :
 Pour I allant de 1 à MPOS(NV+1), MPMG(I)=1.
 Avant le traitement général :
 MPMG(I)=1 si on doit conserver cette modalité.
 MPMG(I)=0 si on la supprime purement et simplement.
 MPMG(I)=-J si on la jumele avec la modalité n° J de la variable.
 Après le traitement général :
 MPMG(I)=0 si on ne conserve pas cette modalité.
 MPMG(I)=K si c'est la k-ième modalité à conserver.
 MPMG(I)=-(100+K) si on la jumele avec la k-ième modalité à conserver. (K dans la numérotation globale)
- MPOSVG : Vecteur des positions des modalités à conserver pour les variables à conserver.
- MPVG : Vecteur indicateur des variables à conserver.
 Au départ :
 MPVG(I)=1 pour I allant de 1 à NV.
 A la fin :
 MPVG(I)=0 si cette variable ne doit pas être conservée.
 MPVG(I)=-1 si cette variable doit être éliminée pour cause de poids trop important de la modalité non-répondue.
 MPVG(I)=K si c'est la K-ième variable à conserver.

NCODI : Vecteur contenant les codages condensés pour un individu.

NICON : Vecteur des nombres d'individus concernés pour les différentes modalités.

PM : Vecteur des poids des modalités.

Concernant les dimensions.

- - - - -

MPMG(NTOTM)
MPOSVG(NV+1)
MPVG(NV)
NCODI(NUMVR(NV))
NICON(NTOTM)
PM(NTOTM)

Sous-programmes appelés.

- - - - -

SAVAL

LECL

Action de NETTOY.

- - - - -

Effet sur l'argument.

ZTOTAL aura la valeur .TRUE. s'il y a eu un problème, soit lors de la ventilation (pas de modalité de poids suffisant), soit lors des regroupements (insuffisance de poids malgré regroupement). Il aura la valeur .FALSE. si tout s'est passé correctement.

Effet sur les variables communes avec PRINCL.

COMMON/COM4/: pas de changement.

COMMON/COM5/:

NV : Nombre de variables conservées.

NI : Nombre d'individus conservés.

PI : Vecteur dont les NI premières composantes sont les poids des individus concernés.

MPOS : Vecteur des positions des modalités des variables, compacté en fonction des variables et des modalités conservées.

NGR : Nombre de groupes conservés.

NVIGR : Vecteur dont les NGR premières composantes sont les nombres d'individus restants dans les groupes conservés.

ILIB, NUMVR, NICLA : inchangés.

COMMON/COM6/:

NUMRES : inchangé

NTOTM : Nombre de modalités conservées.

COMMON/COM7/:

FCOD1 : = FCOD si on a gardé toutes les modalités
Fl.DAT sinon.

COMMON/COM8/: pas de changement.

COMMON/COM9/:

LIBVAR : Si ILIB=ILIB2, les NV premières composantes sont les libellés
des variables conservées.

LIBMOD : Si ILIB=ILIB2, vecteur des modalités des différentes variables
compacté en fonction des variables et des modalités conservées.

Effet sur le fichier de nom FCOD1.

Il contient le tableau C des codages condensés après nettoyage.
C'est le tableau des codages condensés de départ compacté horizontalement
en fonction des variables conservées, et verticalement en fonction des
individus conservés. Tous ses éléments vérifient l'inégalité

$$1 \leq C(I,J) \leq \text{MPOS}(J+1) - \text{MPOS}(J)$$

pour les nouvelles valeurs de MPOS.

VII.1.4 Description de SELECT (SELECT.FOR).

Arguments.

NOUT : Indicateur du choix du mode de sortie pour les résultats intermédiaires.

FRES : Nom du fichier des résultats intermédiaires (si NOUT=NOUT3).

Identificateurs employés pour les variables, tableaux et constantes.

Identificateurs se trouvant dans les listes de COMMON.

Voir PRINCl.

Identificateurs de constantes locales à SELECT.

NUMCC1 : Numéro d'unité pour les accès au fichier FCOD1 .

NUMCCN : " " " " " " " FCODN.

NUMFLN : " " " " " " " FLIBN.

NBURT : " " " " " " " FBURT.

NBURTA : " " " " " " " FBURTA.

NBURTY : " " " " " " " FBURTY.

NFDON2 : " " " " " " " FDON2.

Identificateurs de variables simples.

CTM : Coefficient de Tschuprow multiple au pas précédent.

CTM1 : Nouveau coefficient de Tschuprow multiple.

IVS : Indice de la variable sélectionnée.

NPAS : Numéro de l'itération.

NPER1 : Compteur du nombre de fois qu'il faut ajouter ICTF à 1 pour obtenir le numéro de l'itération où il faudra éditer le tableau des coefficients de Tschuprow entre les variables.

NPER2 : Idem NPER1 mais pour les coefficients de Tschuprow entre les variables explicatives et Y (ICTYF).

NVSEFF : Nombre de variables sélectionnées.

TSCMAX : Coefficient de Tschuprow maximum avec Y.

ZSELEC : Indicateur de continuation de la sélection.

Identificateurs de tableaux.

- BURT : Tableau contenant les tranches successives de la matrice $X^t D_p X$.
C'est-à-dire les matrices $X_i^t D_p X$, $i=1, \dots, NV$
- BURTY : Tableau contenant la matrice $X^t D_p Y$.
- LIBV1 : Vecteur des libellés des variables sélectionnées (C).
- LIBM1 : Vecteur des libellés des modalités des variables sélectionnées (C).
- MPOS1 : Vecteur des positions des modalités des variables sélectionnées.
- MPVS : Vecteur indicateur de l'état des variables.
Initialisation :
MPVS(I)=0 pour I allant de 1 à NV.
A la fin :
MPVS(I)=K si la variable n° I est la K-ième sélectionnée.
MPVS(I)=-(J+NV) si la variable n° I a un coefficient de Tschuprow d'ordre 1 proche de 1 avec la variable J.
MPVS(I)=-J si la variable n° I a un coefficient de Tschuprow d'ordre J proche de 1 avec la J-ième variable sélectionnée.
MPVS(I)=0 dans les autres cas.
- NCODI : Vecteur pour la lecture d'une ligne du fichier des codages condensés FCOD1.
- NCODIN: Vecteur NCODI réordonné en fonction des variables sélectionnées.
- TETAK : Vecteur qui contient les sommes des lignes du tableau de Burt croisant les modalités de deux variables.
- TETAL : Idem TETAK mais pour les sommes des colonnes.
- TSCHX : Vecteur contenant le triangle inférieur (diagonale comprise) de la matrice des coefficients de Tschuprow entre les variables explicatives.
- TSCHY : Vecteur des coefficients de Tschuprow avec Y.

Concernant les dimensions.

- - - - -

BURT (MODMAX)
BURTY (NTOTM,NGR)
LIBM1 (NTOTM)
LIBV1 (NV)
MPOS1 (NV+1)
MPVS (NV)
NCODI (NV)
NCODIN (NV)
TETAK (MODMAX)
TETAL (MODMAX)
TSCHX (NV.(NV+1)/2)
TSCHY (NV)

Sous-programmes appelés.

RINTAB

ROUTAB

Action de SELECT.

Effet sur les arguments : pas de changement.

Effet sur les variables communes : pas de changement.

Création d'un nouveau fichier de codages condensés (FCODN) qui contient les NVSEFF colonnes de FCOD1 relatives aux variables sélectionnées, dans l'ordre de sélection.

Si un fichier de libellés était utilisé, création d'un nouveau fichier ayant la structure correspondant à des libellés de modalités différents pour les variables et dans lequel on trouve les libellés des variables, les nombres de modalités et les libellés des modalités pour chaque variable sélectionnée.

Création du fichier FBURT qui contient la matrice $X^t_{D_p} X$ pour les variables sélectionnées.

Création du fichier FBURTY qui contient la matrice $Y^t_{D_p} X$ pour les variables sélectionnées.

Création du fichier des données pour le second programme, FDON2 qui contient :

```

NVSEFF
MPOS1(1),...,MPOS1(NVSEFF+1)
NI
NICLA
PI(1),...,PI(NI)
NGR
NVIGR(1),...,NVIGR(NGR)
FBURT
FBURTY
FCODN
NOUT
Si NOUT=NOUT3, FRES
VLPMIN
PDISCU.

```


VII.1.5 Description de RINTAB (SELECT.FOR).

Arguments.

TAB : Vecteur dans lequel on stocke le triangle inférieur (diagonale comprise) d'une matrice symétrique.

NDIM : Dimension du vecteur TAB.

I : Indice de la ligne de l'élément à introduire.

J : Indice de la colonne de l'élément à introduire.

VAL : Valeur à introduire.

Action.

Met dans TAB, à l'endroit correspondant aux coordonnées I et J de la matrice, la valeur VAL.

VII.1.6 Description de ROUTAB (SELECT.FOR).

Arguments.

TAB : Idem RINTAB.

NDIM : " "

I : Indice d'une ligne de la matrice.

J : Indice d'une colonne de la matrice.

Effet.

Retourne la valeur de l'élément (I,J) de la matrice stockée dans le vecteur TAB.

VII.1.7 Description de SAVAl, SAVA2, SAVA3, SAVA4, SAVA5, SAVA6, SAVA7, SAVA9, SAVAl0, SAVAl1, SAVAl2, SAVAl3 et SAVAl4.

Voir les indications dans le fichier SAVA.FOR.

VII.1.8 Description de ZVLD1, ZVLD2, ZVLD3, ZVLD4, ZVLD5, ZVLD6, ZVLD7,
ZVLD8, ZVLD9, ZVLD10, ZVLD11, ZVLD12, ZVLD13 et ZVLD14.

Voir les indications dans le fichier VALIDA.FOR.

VII.1.9 Description de LEC1, LEC2, LEC3, LEC4, LEC5, LEC6, LEC7 et LEC8.

Voir les indications dans le fichier LECTUR.FOR.

VII.1.10 Description de MSGERR et de MSGT.

Voir les indications dans le fichier MSG.FOR.

VII.2 Description du deuxième programme.

=====

Rappelons qu'il est chargé d'effectuer l'analyse factorielle discriminante, le reclassement des individus et le classement des individus supplémentaires suivant les méthodes et remarques exposées dans les chapitres précédents. Le programme principal s'appelle PRINC2 et se trouve dans le fichier PRINC2.FOR. Les procédures et les fonctions nécessaires à son exécution sont :

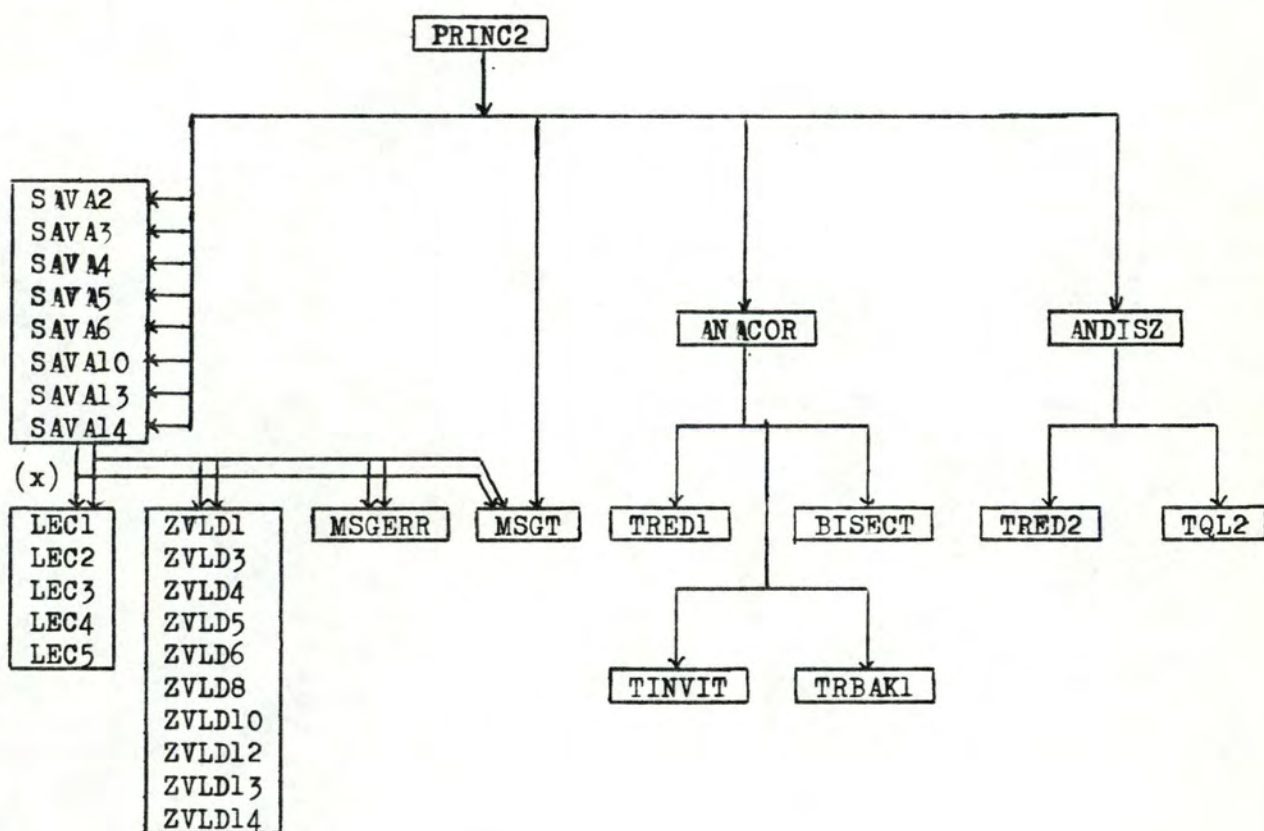
Procédures :

MSGT	
MSGERR	(dans MSG.FOR)
SAVA2	
SAVA3	
SAVA4	
SAVA5	
SAVA6	
SAVA10	
SAVA13	
SAVA14	(dans SAVA.FOR)
LEC1	
LEC2	
LEC3	
LEC4	
LEC5	(dans LECTUR.FOR)
ANACOR	(dans ANACOR.FOR)
ANDISZ	(dans ANDISZ.FOR)
TRED1	
BISECT	
TINVIT	
TRBAK1	
TRED2	
TQL2	(dans la librairie EISDP)

Fonctions :

ZVLD1
ZVLD3
ZVLD4
ZVLD5
ZVLD6
ZVLD8
ZVLD10
ZVLD12
ZVLD13
ZVLD14

Schéma des appels :



(x): Voir **SAVA.FOR** pour le détail des appels.

VII.2.1 Description de PRINC2 (PRINC2.FOR).

Identificateurs employés pour les variables, constantes et tableaux.

NOUT1 :
 NOUT2 :
 NOUT3 : voir PRINC1, COMMON/COM1/

VLPMIN :
 PDISCU : " " , COMMON/COM2/

FBURT :
 FBURTY : " " , COMMON/COM3/
 NUMTER : " " , COMMON/COM4/

NV :
 NI :
 NICLA :
 PI :
 MPOS :
 NGR :
 NVIGR : " " , COMMON/COM5/

NUMRES :
 NTOTM : " " , COMMON/COM6/

FCOD : " " , COMMON/COM8/

FDON :
 FRES :
 NUMFR :
 NUMLPT :
 NOUT : " "

NUMCC :
 NVMIN :
 NVMAX :
 NCLMI :
 NGRMIN :
 NGRMAX :
 PDCINF :
 EPSPDC :
 VLPINF :
 VLPSUP :
 EPSVLP :
 PIINF :
 PISUP :
 EPSPI :
 SOMPI :
 EPSPI :
 NCODI : voir SAISIE.

Identificateurs de variables simples et de constantes non encore rencontrés.

- ID : Indice du vecteur propre de B_Z associé à la première valeur propre supérieure à 1.D-8.
- IERR : Indicateur d'erreur dans ANACOR ou ANDISZ.
- NBGAR : Nombre de facteurs z conservés.
- NBURT : Numéro d'unité pour les accès au fichier FBURT.
- NBURTY : " " " " " " " FBURTY.
- NBVLP : Nombre de valeurs propres trouvées par ANACOR.
- NDIM : Nombre de modalités maximum permis.
- NFDON : Numéro d'unité pour les accès au fichier FDON.
- NFNT : Nombre de facteurs z non triviaux.
- UNV : Borne supérieure de l'intervalle dans lequel rechercher les valeurs propres pour ANACOR.
- ZE : Indicateur d'erreur pour les données générales.
- ZP : Indicateur d'erreur pour la saisie de FDON.

Identificateurs de tableaux.

- A : Contient au départ la matrice $X^t D_p X$.
 Contient avant ANACOR la matrice à diagonaliser.
 Contient avant ANDISZ, la matrice B_Z .
 Contient après ANDISZ, les vecteurs propres de B_Z .
- NBPG : Vecteur qui contiendra pour chaque groupe, le nombre d'individus de ce groupe reclassés dans chaque groupe.
- NTRI : Vecteur indicateur des facteurs z à conserver après le tri.
 $NTRI(I)=0$ si on ne doit pas garder le facteur z n° I.
 $NTRI(I)=K$ si le facteur z n° I est le K-ième à garder.
- PM : Vecteur des poids des modalités.
- PMDEMI : Vecteur des racines carrées des poids des modalités.
- ROT : Vecteur de rotation.
- VALEUR : Vecteur qui, pour un individu, contient les valeurs de $\bar{z}^k \cdot \bar{z}^k - 2 \cdot \bar{z}^k \cdot U^t \cdot e$ pour chaque groupe.
- VLP : Vecteur de valeurs propres.

- VP : Matrice des vecteurs propres de l'analyse des correspondances.
- YTDPY : Vecteur qui contient dans un premier temps les poids des groupes, et, dans un deuxième temps, les produits scalaires des centres de gravité des différents groupes.
- YTDPZ : Tableau qui contiendra successivement les matrices :

$$\begin{aligned}
 & Y^t_{D_p} X \\
 & Y^t_{D_p} Z \\
 & (Y^t_{D_p} Y)^{-1} Y^t_{D_p} Z \\
 & (Y^t_{D_p} Y)^{-1} Y^t_{D_p} Z U^t.
 \end{aligned}$$

Concernant les types.

Voir PRINC1.

Sous-programmes appelés par PRINC2.

SAVA2
SAVA3
SAVA4
SAVA5
SAVA6
SAVA10
SAVA13
SAVA14

MSGT

ANACOR

ANDISZ

Concernant les dimensions des tableaux.

Si vous désirez changer certaines bornes, il faudra toujours veiller à ce que les tableaux aient au moins les dimensions suivantes :

PI, MPOS, NVIGR, PM : voir PRINC1.
A (NTOTM, NTOTM)
NBPG (NGR)
NTRI (NTOTM-NV+1)
PMDEMI (NTOTM)
ROT (NTOTM)
VALEUR (NGR)
VLP (NTOTM-NV+1)

VP (NTOTM,NTOTM-NV+1)
 YTDPY (NGR)
 YTDPZ (NGR,NTOTM)

VII.2.2 Description de SAVA2, SAVA3, SAVA4, SAVA5, SAVA6, SAVAL0, SAVAL3 et SAVAL4.

Voir les indications dans le fichier SAVA.FOR

VII.2.3 Description de LEC1, LEC2, LEC3, LEC4 et LEC5.

Voir les indications dans le fichier LECTUR.FOR

VII.2.4 Description de ZVLD1, ZVLD3, ZVLD4, ZVLD5, ZVLD6, ZVLD8, ZVLD10, ZVLD12, ZVLD13, ZVLD14.

Voir les indications dans le fichier VALIDA.FOR.

VII.2.5 Description de MSGERR et MSGT.

Voir les indications dans le fichier MSG.FOR.

VII.2.6 Description de ANACOR et ANDISZ.

Voir [R8], pages 35, 36, 211 - 219, 448 - 457, 475 - 478 et 483 - 488
 pour ANACOR.

Voir [R8], pages 32, 33, 468 - 474 et 489 - 494 pour ANDISZ.

CHAPITRE VIII :

- APPLICATION -

Le problème qui m'était posé, était le suivant :

A partir de la sensibilité d'un germe donné à un certain nombre d'antibiotiques, pourrait-on identifier ce germe ?

A propos des microbes :

On dispose de données sur 32 microbes (p. VIII.9) qui peuvent être séparés en quatre grandes familles :

- A) Staphylocoques (STA) = coques en amas
- B) Streptocoques (STR) = coques en chaînettes
- C) Entérobactéries (ENT) = bacilles fermentant le glucose
- D) Non-fermentant (NF) = bacilles ne fermentant pas le glucose.

Une première classification serait de considérer qu'il y a quatre groupes.

Combinaison 1 : groupe 1 : STA

groupe 2 : STR

groupe 3 : ENT

groupe 4 : NF

On pourrait aussi grouper les staphylocoques et les streptocoques qui sont des coques et d'autre part, les entéro-bactéries et les non-fermentants qui sont des bacilles.

Combinaison 2 : groupe 1 : STA et STR

groupe 2 : ENT et NF

Deux autres combinaisons possibles seraient :

Combinaison 3 : groupe 1 : STA et STR

groupe 2 : ENT

groupe 3 : NF

Combinaison 4 : groupe 1 : STA
 groupe 2 : STR
 groupe 3 : ENT et NF.

A propos de la sensibilité, de la résistance et des antibiotiques.

Le laboratoire considère trois grands niveaux :

Sensible (S)
 Résistant (R)
 Intermédiaires (I).

Les tableaux envisagent en plus :

S(R) : La plupart des microbes sont sensibles, certains sont résistants.

S/I : Les microbes sont, soit sensibles, soit intermédiaires.

VAR : Les microbes sont soit sensibles, soit résistants, soit inter-
 médiaires.

R/I : Les microbes sont soit résistants, soit intermédiaires.

R(S) : La plupart des microbes sont résistants; certains sont sensibles.

On dispose de 39 antibiotiques (p. VIII.10), mais il faut également faire l'analyse sur une partie d'entre elles (27 exactement, dont les numéros sont : 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 14, 15, 18, 20, 21, 22, 24, 26, 27, 28, 30, 31, 33, 34, 35 et 36).

A propos des poids des microbes et des modalités.

Le nombre de données dont on dispose est très faible. De plus, il y a un déséquilibre dans les nombres d'individus (2 STA, 2 STR, 14 ENT, 14 NF). C'est pourquoi j'ai décidé de ne pas attribuer le même poids à chaque individu. J'ai donné aux coques le poids 0.11 et aux bacilles le poids 0.02. On obtient ainsi, pour les différentes combinaisons :

Combinaison 1 : groupe 1 : .22

groupe 2 : .22

groupe 3 : .28

groupe 4 : .28

Combinaison 2 : groupe 1 : .44

groupe 2 : .56

Combinaison 3 : groupe 1 : .44

groupe 2 : .28

groupe 3 : .28

Combinaison 4 : groupe 1 : .22

groupe 2 : .22

groupe 3 : .56

Toutes les modalités de poids non nul sont de poids suffisant, et, j'ai fixé le poids maximum autorisé pour la modalité non-répondu à .15.

Traitement du premier cas : combinaison STA, STR, ENT, NF.

Nous avons au départ 39 variables totalisant 351 modalités. Après nettoyage, c'est-à-dire dans ce cas-ci, élimination des modalités de poids nul et des variables dont le nombre de modalités est égal à 1, ou dont le poids de la modalité non-répondu est trop important (cela concerne les variables 6, 7, 9, 18, 24, 37 et 39), il reste 32 variables et 167 modalités. La sélection de 10 variables fournit le tableau 1 où :

IV = indice de la variable sélectionnée

CTMAX = coefficient de Tschuprow maximum avec Y.

CTM = coefficient de Tschuprow multiple.

CROISS= croissance du coefficient de Tschuprow multiple
par rapport à l'étape précédente.

NBCM = nombre cumulé de modalités.

	IV	CTMAX	CTM	CROISS	NBCM
Pas 1 : 36	.6455	.64553	.64553	4	
Pas 2 : 27	.3716	.70516	.059626	10	
Pas 3 : 2	.2523	.72749	.022339	14	
Pas 4 : 20	.2137	.74213	.014639	17	
Pas 5 : 32	.1995	.75408	.011970	21	
Pas 6 : 35	.1668	.76201	.007924	25	
Pas 7 : 14	.1245	.76626	.004256	31	
Pas 8 : 31	.1117	.76962	.003356	36	
Pas 9 : 13	.0991	.77222	.002599	42	
Pas 10 : 5	.0786	.77383	.001616	48	

- Tableau 1 -

Si on effectue l'analyse discriminante par nombre croissant de variables, on obtient :

	IV	NFNT	PDISCU	Reclassement des individus de départ				Pourcentage total de biens classés
				Gr 1	Gr 2	Gr 3	Gr 4	
1 var. :	36	3	1.9365	2/2	2/2	13/14	3/14	62.5
2 var. :	27	8	2.5608	2/2	2/2	13/14	11/14	87.5
3 var. :	2	11	2.7510	2/2	2/2	13/14	13/14	93.75
4 var. :	20	13	2.8836	2/2	2/2	13/14	14/14	96.875
5 var. :	32	14	2.8843	2/2	2/2	13/14	14/14	96.875
6 var. :	35	17	2.9142	2/2	2/2	14/14	14/14	100.00

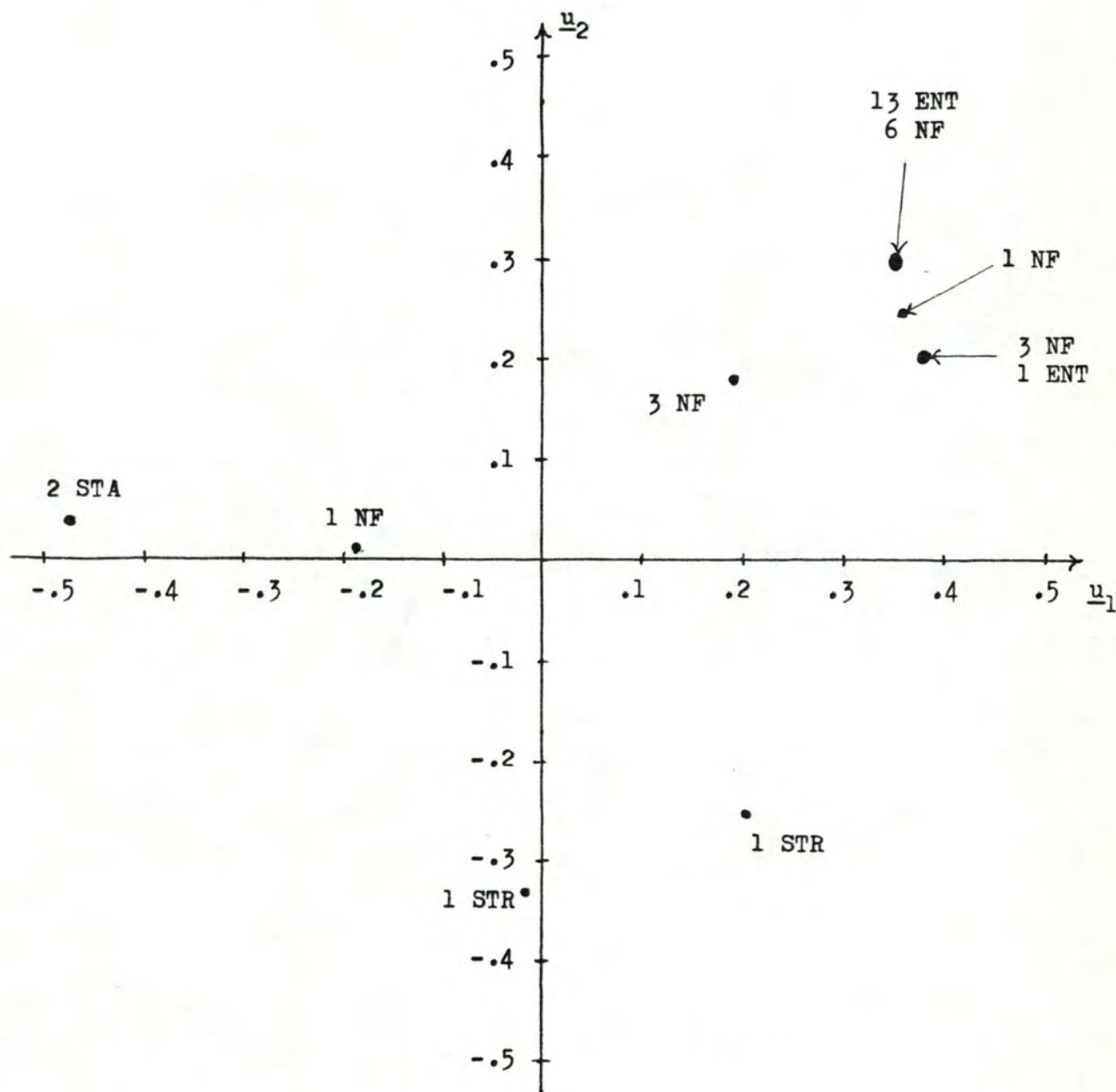
- Tableau 2 -

NFNT = Nombre de facteurs z non triviaux conservés

PDISCU = Pouvoir discriminant cumulé pour les facteurs z conservés.

6 variables (25 modalités) suffisent donc pour reclasser convenablement tous les germes.

Si on représente les individus dans le plan déterminé par les deux premiers codages discriminants (voir p. VIII.11), on obtient la figure suivante :



- figure 1 -

C'est l'entérobactérie isolée qui nous a forcé à considérer 6 variables (au lieu de 4) pour pouvoir reclasser correctement tous les microbes. Ce graphique est très intéressant car il nous montre que s'il sera très facile de séparer les staphylocoques du reste, ainsi que les streptocoques,

la discrimination entre bacilles ne sera pas aisée.

Si on se limite aux 27 variables mentionnées page VIII.2, les résultats sont tout à fait du même ordre de grandeur et le reclassement suit exactement la même progression. (Les variables sélectionnées sont les variables 36, 27, 2, 20, 14 et 35).

Traitement du deuxième cas : combinaison STA et STR, ENT et NF.

Comme prévu, un petit nombre de variables sont nécessaires pour reclasser correctement les individus. Avec 2 variables, on obtient :

IV	CTMAX	NFNT	PDISCU	Reclassement	% total
1 var. : 36	.5343	3	.9255	4/4	27/28 96.875
2 var. : 34	.4138	6	.96213	4/4	28/28 100.00

- Tableau 3 -

Les résultats sont identiques pour le sous-ensemble de 27 antibiotiques.

Traitement du troisième cas : combinaison STA et STR, ENT, NF.

5 variables (34, 35, 36, 27 et 22) suffisent pour reclasser tous les individus sauf un non-fermentant (le deuxième) et 10 variables n'améliorent pas la situation.

Traitement du quatrième cas : combinaison STA, STR, ENT et NF.

Comme prévu, tout se passe sans problème et les 6 facteurs z issus de l'analyse des correspondances des variables 36 et 2 ont un pouvoir discriminant cumulé égal à 2, ce qui est le maximum pour 3 groupes. Il n'y a donc rien d'étonnant à ce que ces deux variables permettent de reclasser correctement tous les individus.

Affectation d'individus supplémentaires.

Malgré que le nombre d'observations soit trop peu important pour extraire un échantillon test et en classer les individus, j'ai néanmoins essayé dans le cas de la combinaison 2. . .

L'échantillon test était composé des

- coques n° 4
- bacilles n° 1, 2, 3, 26, 27 et 28.

Dans ce cas ci, l'analyse discriminante sur les observations restantes a été encore plus rapide que pour le deuxième cas puisque la variable 32 à elle seule a permis de reclasser correctement tous les individus de départ.

Le classement des individus supplémentaires a donné les résultats suivants :

- Pour une variable sélectionnée (32), ils sont tous bien classés, excepté le bacille 27 qui est classé dans les coques, ce à quoi il fallait s'attendre au vu de sa position (le NF le plus à gauche) sur la figure 1.
- Pour deux variables, il en est de même.
- Pour trois variables (32, 2 et 1), le coque n° 4 va se classer parmi les bacilles. Dans ce cas ci, il ne faut donc certainement pas plus de deux variables pour reclasser des individus supplémentaires.

J'ai également tenté l'expérience pour l'échantillon test suivant :

- coques n° 1 et 4
- bacilles n° 1, 2, 3, 12, 13, 14, 15, 16, 17, 26, 27 et 28.

Une seule variable (32) a suffi à reclasser les 18 individus constituant l'échantillon de base.

Les individus de l'échantillon test ont tous été bien classés, sauf les bacilles 17, 27 et 28, qui correspondent en fait aux NF 3,13 et 14, c'est-à-dire aux 3 parmi les 4 NF les plus à gauche sur la figure 1. L'ajout de variables supplémentaires n'y a rien changé.

Conclusions de ces expériences.

Bien qu'il faille rester très prudent quant aux conclusions à tirer de ces analyses, étant donné le petit nombre d'observations, il me semble que l'on peut quand même affirmer que :

- 1) Un petit nombre d'antibiotiques permettent de discriminer de façon relativement satisfaisante entre les microbes.
- 2) Si un germe inconnu est classé dans les bacilles, il y a de très fortes chances pour que ce soit bien un bacille.
- 3) Si un germe inconnu, mais dont on sait qu'il n'est pas un NF, est classé dans les coques, la probabilité que ce soit un coque est également élevée.

Temps d'exécution :

Le temps d'exécution du premier programme ne dépend pas tellement du nombre de variables à sélectionner, mais plutôt du nombre de modalités conservées après nettoyage. Ce qui prend du temps, c'est la construction du tableau de Burt et le calcul des coefficients de Tschuprow du premier ordre. Pour les applications précédentes, les temps ont été de l'ordre de 1'15" pour 39 variables et de 50" pour 27.

L'exécution du second programme n'a jamais pris plus de 35".

Liste des microbes.

=====

- 1) Staph. aureus
- 2) Staph. epidermidis
- 3) Streptococci sp.
- 4) Streptococcus faecalis
- 5) E. coli
- 6) Shigella sp.
- 7) Citrobacter diversus (Koseri)
- 8) Citrobacter freundii
- 9) Salmonella sp.
- 10) Klebsiella pneum.
- 11) Klebsiella oxytoca
- 12) Enterobact aerogen.
- 13) Enterobact cloacae
- 14) Serratia marcese
- 15) Proteus mirabilis
- 16) Proteus vulgaris
- 17) Morg. morganii
- 18) P. rettgeri
- 19) Providencia st.
- 20) Yersinia enterocolitica
- 21) Yersinia pseudotuberc.
- 22) Pseud. aeruginosa
- 23) Pseud. fluorescens-putida
- 24) Pseud. cepacia
- 25) Pseud. maltophilia
- 26) Pseud. stutzeri
- 27) Alcaligenes faecalis
- 28) Bordetella bronchisept.
- 29) Acinetobact. amitratum
- 30) Acinetobact. lwoffii
- 31) Moraxella sp.
- 32) Flavobact. meningosept.

Liste des antibiotiques.

- =====
- 1) Pen G
 - 2) Methicillin
 - 3) Ampicillin
 - 4) Carbenicillin (Ticarcillin)
 - 5) Mecillinam
 - 6) Azlocillin
 - 7) Mezlocillin
 - 8) Piperacillin
 - 9) 1st gen. cephalosporins (Cephalothin)
 - 10) Cefamandole
 - 11) Cefuroxime
 - 12) Cefoxitin
 - 13) Cefsulodin
 - 14) Cefotaxime
 - 15) Moxalactam
 - 16) Ceftazidim
 - 17) Streptomycin
 - 18) Kanamycin
 - 19) Neomycin
 - 20) Gentamycin
 - 21) Tobramycin
 - 22) Amikacin
 - 23) Netilmicin
 - 24) Tetracycline
 - 25) Minocycline
 - 26) Chloramphenicol
 - 27) Erythromycin
 - 28) Clindamycin
 - 29) Sulphonamides
 - 30) Trimethoprim
 - 31) Trimethoprim Sulfa.
 - 32) Bacitracin
 - 33) Polymyxins (Colistin)
 - 34) Vancomycin
 - 35) Nitrofurantoin
 - 36) Nalidixic acid
 - 37) Fusidic acid
 - 38) Novobiocin
 - 39) Rifampicin

Codages discriminants pour le premier cas.

Codage 1	Codage 2	Codage 3
-----	-----	-----
-0.0503	0.2595	-0.5200
0.1784	-0.9199	-1.2701
-0.0503	0.2595	1.7909
-0.0503	0.2595	1.1107
0.0	0.0	0.9668
0.0	0.0	-1.9702
0.0	0.0	1.4961
0.0	0.0	-0.0896
0.0	0.0	1.3519
0.0	0.0	0.7258
0.5938	-0.8238	-0.4417
0.2908	0.0672	0.5798
-1.2678	0.1697	-1.7211
1.2292	0.4799	2.5167
0.0	0.0	-0.4776
0.0	0.0	1.8334
0.0	0.0	-1.2479
-0.5400	-0.1248	-1.0768
0.3983	0.2879	1.1682
-0.2370	-1.0159	-2.0985
0.3983	0.2879	0.7831
0.0	0.0	-0.0114
0.0	0.0	0.3065
0.0	0.0	-0.1777
0.0	0.0	-0.5606

- ANNEXES -

ANNEXE A : LE COEFFICIENT DE TSCHUPROW COMME COSINUS D'ANGLE

=====

DANS UN ESPACE D'OPERATEURS.

Soit W_i , une variété linéaire engendrée par une matrice d'indicateurs X_i de dimension $N \times p$

$$W_i = \{ \underline{w} \text{ t.q. } \underline{w} = X_i \cdot \underline{a}, \underline{a} \in R^p \}.$$

La matrice de projection D_p - orthogonale sur W_i se construit très facilement en tenant compte du fait que, pour tout x de R^N , $\underline{x} - P_i \cdot \underline{x}$ doit être D_p - orthogonal à tout vecteur \underline{w} de W_i si \underline{x} n'appartient pas à W_i . On en déduit immédiatement l'expression de P_i :

$$P_i = X_i \cdot (X_i^t \cdot D_p \cdot X_i)^{-1} \cdot X_i^t \cdot D_p$$

P_i est une matrice $N \times N$ et donc un élément d'un espace d'opérateurs de R^N dans R^N . Si on considère, sur cet espace, le produit scalaire de la trace du produit, $\langle A; B \rangle = \text{trace} (A \cdot B)$ et le cosinus qui lui est associé

$$\cos (A; B) = \frac{\langle A; B \rangle}{\sqrt{\langle A; A \rangle \cdot \langle B; B \rangle}} = \frac{\text{tr}(A \cdot B)}{\sqrt{\text{tr}(A^2) \cdot \text{tr}(B^2)}},$$

on peut montrer que, si P_i et P_j sont les projecteurs D_p - orthogonaux respectivement sur W_i et W_j ,

$$\text{tr} (P_i \cdot P_j) = 1 + \text{PHI}_{ij}^2$$

$$\text{tr} (P_i^2) = m_i, \text{tr} (P_j^2) = m_j$$

et de ce fait

$$\cos (P_i; P_j) = \frac{1 + \text{PHI}_{ij}^2}{\sqrt{m_i \cdot m_j}}.$$

Si maintenant, au lieu des W_i , nous considérons les sous-espaces des variables centrées, W_{i0} et les projecteurs D_p - orthogonaux associés, P_{i0} ,

nous aurons

$$\text{tr} (P_{i0} . P_{j0}) = \text{PHI}_{ij}^2$$

$$\text{tr} (P_{i0}^2) = m_i - 1 , \text{tr} (P_{j0}^2) = m_j - 1$$

et ainsi, le cosinus entre les opérateurs P_{i0} et P_{j0} sera donné par

$$\cos (P_{i0}, P_{j0}) = \frac{\text{PHI}_{ij}^2}{\sqrt{(m_i - 1) \cdot (m_j - 1)}}$$

qui est exactement l'expression du coefficient de Tschuprow entre les variables χ_i et χ_j .

ANNEXE B : ANALYSE CONJOINTE DE P VARIABLES QUALITATIVES.

=====

L'analyse conjointe de p variables qualitatives χ_i , $i=1, \dots, p$ revenant à effectuer l'analyse conjointe des p tableaux d'indicatrices, voyons comment effectuer celle-ci dans le cas plus général de p tableaux rectangulaires X_i de dimension $N \times m_i$, représentant chacun les observations de m_i variables numériques sur N individus de poids p_i , $\sum p_i = 1$. Nous supposons que les variables sont centrées.

B.1 Analyse conjointe de p tableaux de variables numériques.

Dans le cas de deux ensembles de variables, l'analyse des liaisons existant entre ces groupes s'effectue au moyen de l'analyse canonique ordinaire qui détermine des couples (w_1, w_2) de combinaisons linéaires de variables de chaque groupe les plus corrélées entre elles. Généraliser l'analyse canonique à plus de deux ensembles, débouche sur la difficulté de mesurer la dépendance entre plus de deux variables. En effet, si on veut déterminer un p-uple de variables canoniques D_p - normées (w_1, \dots, w_p) , $w_1 \in W_1$, le choix du critère à maximiser n'est pas indifférent, or il en existe plusieurs qui redonnent l'analyse canonique habituelle lorsque $p = 2$.

Une méthode pour déterminer ce p-uple de variables canoniques est de rechercher une variable auxiliaire unique la plus liée en un certain sens aux ensembles X_i et de la décomposer ensuite sur les W_i pour obtenir les w_i . On recherche donc une variable unique z , la plus liée en moyenne aux p ensembles de variables au sens où la somme des carrés des coefficients de corrélation multiple de z avec les X_i soit maximale, et on prendra pour w_i la projection D_p -orthogonale de z sur W_i .

Le carré du coefficient de corrélation multiple R_i de z avec X_i n'étant autre que le carré du cosinus de l'angle formé par z et W_i , c'est-à-dire, par z et sa projection D_p - orthogonale sur W_i , on a

$$R_i^2 = \frac{z^t \cdot D_p \cdot P_i \cdot z}{z^t \cdot D_p \cdot z}.$$

Il suffit donc de trouver une variable z qui maximise $\sum_{i=1}^p R_i^2$; c'est-à-dire solution du problème

$$\max \frac{z^t \cdot D_p \cdot \left(\sum_{i=1}^p P_i \right) \cdot z}{z^t \cdot D_p \cdot z}.$$

La solution z est le vecteur propre de $\sum P_i$ associé à sa plus grande valeur propre f_1 qui est alors la valeur maximale de $\sum R_i^2$.

Ayant obtenu un premier facteur z , on peut poursuivre la recherche des éléments propres de $\sum P_i$ pour obtenir des facteurs D_p - orthogonaux au premier et entre eux.

B.2 Solution dans R^m .

La diagonalisation de $\sum P_i$ de dimension $N \times N$ pose des problèmes dès que le nombre d'observations devient important. C'est pourquoi, on se ramène très simplement à la diagonalisation d'une matrice $m \times m$ en posant $w_i = P_i \cdot z = X_i \cdot b_i$ avec $b_i \in R^m$. On a

$$f \cdot z = \sum_{j=1}^p X_j \cdot b_j$$

d'où

$$P_i \cdot \left(\sum_{j=1}^p X_j \cdot b_j \right) = f \cdot X_i \cdot b_i$$

soit :

$$\left(X_i^t \cdot D_p \cdot X_i \right)^{-1} \cdot X_i^t \cdot D_p \cdot \left(\sum X_j \cdot b_j \right) = f \cdot b_i.$$

En posant $b^t = (b_1^t, \dots, b_p^t)^t$, on obtient que b est vecteur propre associé à la valeur propre f de la matrice $M = D^{-1} \cdot H$ où D est la matrice diagonale

des blocs $X_i^t \cdot D_p \cdot X_i$ tandis que H est la matrice des p^2 blocs $X_i^t \cdot D_p \cdot X_j$.

B.3 Interprétation dans le cas de variables qualitatives.

Pour des variables qualitatives, les données ne sont pas centrées car d'une part, centrer des indicatrices est dénué d'intérêt et d'autre part, les analyses centrées et non centrées conduisent aux mêmes résultats au facteur trivial 1 près. Comme on ne change pas l'analyse en divisant les termes de X par \sqrt{p} , b sera vecteur propre de $D^{-1} \cdot X^t \cdot D_p \cdot X \cdot \frac{1}{p}$ qui est le produit de la matrice $\frac{D_p \cdot X}{p}$ dans laquelle on a divisé chaque élément par le total de sa ligne, par sa transposée où on a divisé chaque élément par le total de sa colonne. La méthode exposée en B.1 appliquée à des variables qualitatives revient donc à effectuer l'analyse des correspondances du tableau X.

ANNEXE C : MODE D'EMPLOI DES PROGRAMMES.

=====

C.1 Pour exécuter le premier programme.

=====

Il suffit de taper la commande :

EXECUTE PRINCl.FOR, SAISIE.FOR, NETTOY.FOR, SELECT.FOR, SAVA.FOR,
VALIDA.FOR, LECTUR.FOR, MSG.FORSi tout se passe correctement au niveau de la compilation et du linkage,
l'ordinateur enverra le message :

```

XXXXXXXXXXXXXXXXXXXXXXX
x METHODE DE SAPORTA. x
x PROGRAMME NUMERO 1. x
XXXXXXXXXXXXXXXXXXXXXXX

```

SI VOUS INTRODUISEZ LES DONNEES AU TERMINAL, TAPEZ 1, SINON TAPEZ 0

Il faut répondre à ce message par 0 ou 1.

Si on répond par 0, le message suivant apparaîtra :

QUEL EST LE NOM DU FICHIER DES DONNES ?

Auquel il faudra répondre en donnant le nom, FDON du fichier des données,
sur une ligne; les 10 premiers caractères représentent la valeur de FDON.
L'ordinateur continuera tout seul.

Si on répond par 1, l'ordinateur posera une série de questions auxquelles
il faudra répondre en respectant les contraintes sur les données générales.
On peut trouver à la fin de l'annexe C un exemple d'introduction de données.

Données générales.

D.1) Le nombre total de variables explicatives à considérer, NV.

Contrainte : $1 \leq NV \leq 100$

Format : un entier sur une ligne.

D.2) Le mode de répartition des variables, MODVAR.

Contrainte : MODVAR = 1 ou MODVAR = 0.

Format : idem D.1

Emploi : Si les variables sont les NV premières dans le fichier des codages condensés, MODVAR doit valoir 1; sinon, MODVAR doit valoir 0.

D.3) Les numéros des variables à considérer, NUMVR(1),...,NUMVR(NV).

Contrainte : D.3 n'est donné que si MODVAR prend la valeur 0.

Pour I allant de 1 à NV :

$$1 \leq \text{NUMVR}(I) \leq 100$$

Si $1 \leq I, J \leq 100$ et $I < J$ alors $\text{NUMVR}(I) < \text{NUMVR}(J)$.

Format : NV entiers sur une même ligne.

D.4) Le nombre total d'individus de base, NI.

Contrainte : $2 \leq NI \leq 4000$

Format : idem D.1

D.5) Le nombre d'individus supplémentaires à classer, NICLA.

Contrainte : $\text{NICLA} \geq 0$

Format : idem D.1

D.6) Le nom du fichier des poids des individus, FPDS.

Contrainte : Si un fichier de nom FDON est utilisé pour l'introduction des données générales, $\text{FPDS} \neq \text{FDON}$.

Format : sur une ligne, les 10 premiers caractères représentent le nom du fichier.

D.7) L'indicateur de l'existence d'un fichier de libellés, ILIB.

Contrainte : $ILIB = 0$ ou $ILIB = 1$

Format : idem D.1

Emploi : Si un fichier de libellés est utilisé, il faut donner à ILIB la valeur 1 et sinon la valeur 0.

D.8) Le nom du fichier des libellés, FLIB.

Contrainte : D.8 n'est donné que si ILIB prend la valeur 1.

Si un fichier de nom FDON est utilisé pour les données générales, $FLIB \neq FDON$. Format : idem D.6

D.9) L'indicateur de l'égalité des libellés de modalités, NELIB.

Contrainte : D.9 n'est donné que si ILIB prend la valeur 1

$NELIB = 0$ ou $NELIB = 1$.

Format : idem D.1

Emploi : Si les libellés des modalités sont les mêmes pour toutes les variables, il faut donner à NELIB la valeur 1 et sinon, la valeur 0.

D.10) L'indicateur de l'égalité des nombres de modalités, NEMOD.

Contrainte : D.10 n'est donné que si ILIB prend la valeur 0.

$NEMOD = 0$ ou $NEMOD = 1$

Format : idem D.1

Emploi : Si toutes les variables ont le même nombre de modalités, il faut donner à NEMOD la valeur 1 et sinon la valeur 0.

D.11) Le nombre de modalités, NMOD (le même pour chaque variable).

Contrainte : D.11 n'est donné que si ILIB prend la valeur 0 et NEMOD la valeur 1.

$1 \leq NMOD \leq 9$

Format : idem D.1

D.12) Les nombres de modalités des variables, $NVMOD(1), \dots, NVMOD(NV)$.

Contrainte : D.12 n'est donné que si ILIB prend la valeur 0 et NEMOD la valeur 0.

Pour I allant de 1 à NV :

$$1 \leq \text{NVMOD}(I) \leq 9$$

Format : idem D.3

D.13) Le nom du fichier des codages condensés, FCOD.

Contrainte : Si un fichier de nom FDON est utilisé pour les données générales, $\text{FCOD} \neq \text{FDON}$.

FCOD est également différent de 'F1.DAT' et 'F2.DAT'

Format : idem D.6

Remarque : 'F1.DAT' et 'F2.DAT' sont deux noms de fichiers qui vont être créés pour stocker le tableau des codages condensés dans ses états intermédiaires. Il faudra donc veiller à ne pas avoir de fichier avec un de ces noms.

D.14) Le nombre de groupes d'individus, NGR.

Contrainte : $2 \leq \text{NGR} \leq 9$

Format : idem D.1

D.15) La liste des nombres d'individus dans chaque groupe, NVIGR(1), ..., NVIGR(N)

Contrainte : Pour I allant de 1 à NGR :

$$1 \leq \text{NVIGR}(I) < \text{NI}$$

La somme pour I allant de 1 à NGR des NVIGR(I) doit valoir NI.

Format : NGR entiers sur une même ligne.

D.16) L'indicateur du choix de la méthode d'arrêt pour la sélection, IMAR.

Contrainte : $\text{IMAR} = 0$ ou $\text{IMAR} = 1$

Format : idem D.1

Emploi : Si on fixe à l'avance le nombre de variables à sélectionner, il faut donner à IMAR la valeur 0, et sinon, la valeur 1.

D.17) La quantité fixant la croissance minimale pour le coefficient de

Tschuprow multiple, QTM

Contrainte : $0 < \text{QTM} < 1$

D.17 n'est donné que si IMAR prend la valeur 1.

Format : Un nombre double précision sur une ligne.

D.18) Le nombre de variables à sélectionner, NVS.

Contrainte : D.18 n'est donné que si IMAR prend la valeur 0.

$$1 \leq NVS \leq NV.$$

Format : idem D.1

D.19) Le poids maximum autorisé pour la modalité "non-répondu" d'une variable, PNRV.

Contrainte : $0 \leq PNRV \leq 1$.

Format : idem D.17

D.20) Le poids minimum pour qu'une modalité soit conservée, PMMIN.

Contrainte : $0 < PMMIN < 1$

Format : idem D.17

D.21) L'indicateur de "quand" on choisit la méthode pour boucher les trous dans le tableau des codages condensés, IQDBT.

Contrainte : $IQDBT = 0$ ou $IQDBT = 1$.

Format : idem D.1

Emploi : Si on veut choisir la méthode avant le nettoyage, il faut donner à IQDBT la valeur 1 et sinon, la valeur 0.

D.22) L'indicateur du choix de la méthode, ICHTR.

Contrainte : D.22 n'est donné que si IQDBT prend la valeur 1.

$$ICHTR = 1 \text{ ou } ICHTR = 2 \text{ ou } ICHTR = 3.$$

Format : idem D.1

Emploi : Si on choisit la méthode par suppression des individus concernés, il faut donner la valeur 1 à ICHTR.

Si on choisit la méthode par ventilation, on lui donnera la valeur 2 et si on choisit celle par regroupement, ce sera la valeur 3.

D.23) L'indicateur du type de regroupement, ITPGR.

Contrainte : D.23 n'est donné que si IQDBT prend la valeur 1 et ICHTR la valeur 3.

ITPGR = 0 ou ITPGR = 1

Format : idem D.1

Emploi : Si les modalités groupes des modalités des variables sont les mêmes pour chaque variable, il faut donner à ITPGR la valeur 1 et sinon, la valeur 0.

D.24) Le vecteur des modalités groupes des modalités d'une variable, NREGR(1), ..., NREGR(NBMOD)

Contrainte : D.24 n'est donné que si IQDBT prend la valeur 1, ICHTR la valeur 3 et ITPGR la valeur 1.

Pour I allant de 1 à NBMOD :

$$1 \leq \text{NREGR}(I) \leq \text{NBMOD}$$

Si $I = \text{NREGR}(K)$ avec $K = I$, alors $\text{NREGR}(I) = I$.

Format : NBMOD entiers sur une même ligne.

D.25) Les vecteurs des modalités groupes des modalités de chaque variable.

Contrainte : D.25 n'est donné que si IQDBT prend la valeur 1, ICHTR la valeur 3 et ITPGR la valeur 0.

Pour chaque vecteur, et pour I allant de 1 à NVMOD(L)

(L étant l'indice de la variable à laquelle correspond le vecteur) :

$$1 \leq \text{NREGR}(I) \leq \text{NVMOD}(L)$$

Si $I = \text{NREGR}(K)$ avec $K = I$, alors $\text{NREGR}(I) = I$.

Format : Un vecteur par ligne suivant le format de D.24

D.26) Le seuil de redondance, SR.

Contraintes : $0.5 < \text{SR} < 1$

Format : idem D.17

D.27 Le nom du fichier des codages condensés après sélection, FCODN.

Contrainte : FCODN doit être différent de FCOD et de 'Fl.DAT'.

FCODN se termine par le suffixe .DAT

Format : idem D.6

D.28) Le nom du fichier des libellés après sélection, FLIBN.

Contrainte : D.28 n'est donné que si ILIB prend la valeur 1

FLIBN \neq FCODN

FCODN se termine par le suffixe .DAT

Format : idem D.6

D.29) Le nom du fichier devant contenir le tableau de Burt croisant les modalités des variables explicatives avant la sélection, FBURTA.

Contrainte : FBURTA \neq FCODN, FLIBN, 'Fl.DAT', FCOD

FBURTA se termine par le suffixe .DAT

Format : idem D.6

D.30) Le nom du fichier devant contenir le tableau de Burt croisant les modalités des variables explicatives après la sélection, FBURT.

Contrainte : FBURT \neq FCODN, FLIBN, FBURTA

FBURT se termine par le suffixe .DAT

Format : idem D.6

D.31) Le nom du fichier devant contenir le tableau de Burt croisant les modalités des variables explicatives avec celles de Y après la sélection, FBURTY.

Contrainte : FBURTY \neq FCODN, FLIBN, FBURT, FBURTA

FBURTY se termine par le suffixe .DAT

D.32) Le nom du fichier devant contenir les données pour le second programme, FDON2.

Contrainte : FDON2 \neq FCODN, FLIBN, FBURT, FBURTA, FBURTY

FDON2 se termine par le suffixe .DAT

Format : idem D.6

D.33) L'inertie minimale pour qu'un facteur z soit conservé, VLPMIN

Contrainte : $0 < \text{VLPMIN} < 1$

Format : idem D.17

D.34) Le pouvoir discriminant cumulé à atteindre pour les facteurs z , PDISCU.

Contrainte : $0 \leq \text{PDISCU} \leq \text{NGR}-1$

Format : idem D.17

D.35) L'indicateur du choix du mode de sortie des résultats, NOUT.

Contrainte : $\text{NOUT} = 1$ ou $\text{NOUT} = 2$ ou $\text{NOUT} = 3$.

Format : idem D.1

Emploi : Si on veut que les résultats intermédiaires apparaissent au terminal, il faut donner à NOUT la valeur 1. Pour les avoir sur imprimante, ce sera la valeur 2 et sur fichier, la valeur 3.

D.36) Le nom du fichier des résultats intermédiaires, FRES.

Contrainte : D.36 n'est donné que si NOUT prend la valeur 3.

$\text{FRES} \neq \text{FCODN}, \text{FLIBN}, \text{FBURT}, \text{FBURTA}, \text{FBURTY}, \text{'Fl.DAT'}, \text{FCOD}.$

FRES se termine par le suffixe .DAT

Format : idem D.6

D.37) L'indicateur d'édition des poids des modalités après nettoyage, IPMN

Contrainte : $\text{IPMN} = 0$ ou $\text{IPMN} = 1$

Format : idem D.1

Emploi : Si on désire la liste des poids des modalités après nettoyage, il faut donner à IPMN la valeur 1 et sinon, la valeur 0.

D.38) L'indicateur d'édition du tableau des coefficients de Tschuprow du premier ordre entre les variables, ICT1.

Contrainte : $\text{ICT1} = 0$ ou $\text{ICT1} = 1$

Format : idem D.1

Emploi : Si on désire ce tableau, il faut donner à ICT1 la valeur 1 et sinon, la valeur 0.

D.39) La période avec laquelle on désire avoir le tableau des coefficients de Tschuprow entre les variables, ICTF.

Contrainte : $ICTF \geq 0$

Format : idem D.1

Emploi : Si on désire avoir le tableau des coefficients de Tschuprow d'ordre k entre les variables, avec $k=1+i.F$, $i=1, \dots$, il faut donner à ICTF la valeur F . Si on ne veut pas l'avoir, il suffit de donner à ICTF la valeur 0.

D.40) L'indicateur d'édition du vecteur des coefficients de Tschuprow d'ordre 1 avec Y, ICTY1.

Contrainte : $ICTY1 = 0$ ou $ICTY1 = 1$

Format : idem D.1

Emploi : Si on désire ce vecteur, il faut donner à ICTY1 la valeur 1 et sinon, la valeur 0.

D.41) La période avec laquelle on désire avoir le vecteur des coefficients de Tschuprow avec Y, ICTYF.

Contrainte : $ICTYF \geq 0$

Format : idem D.1

Emploi : Si on désire avoir le tableau des coefficients de Tschuprow d'ordre k avec Y, $k = 1+i.F$, $i=1, 2, \dots$, il faut donner à ICTYF la valeur F . Si on ne veut pas l'avoir, il suffit de donner à ICTYF la valeur 0.

D.42) L'indicateur d'édition du coefficient de Tschuprow maximum avec Y à chaque étape, ICTMX.

Contrainte : $ICTMX = 0$ ou $ICTMX = 1$

Format idem D.1

Emploi : Si on désire ce coefficient à chaque étape, il faut donner à ICTMX la valeur 1 et sinon, la valeur 0.

D.43) L'indicateur d'édition du coefficient de Tchuprow multiple et de sa croissance à chaque étape, ICTMP.

Contrainte : ICTMP = 0 ou ICTMP = 1

Format : idem D.1

Emploi : Si on désire ces deux choses à chaque étape, il faut donner à ICTMP la valeur 1 et sinon, la valeur 0.

Le fichier des poids des individus.

Il contient les poids des individus, $PI(1), \dots, PI(NI)$.

Le nom de ce fichier est donné par D.6.

Pour I allant de 1 à NI, $0 < PI(I) < 1$.

La somme des $PI(I)$ pour I allant de 1 à NI doit valoir 1.

Les poids des individus se trouvent sur la première ligne, les uns à la suite des autres.

Le fichier des codages condensés.

Il contient le tableau des codages condensés, A.

Le nom de ce fichier est donné par D.13.

A contient au moins NI+NICLA lignes.

A contient au moins NUMVR(NV) colonnes.

Pour I allant de 1 à NI+NICLA et pour J valant successivement NUMVR(1), ..., NUMVR(NV), on a $1 \leq A(I, J) \leq M$ où M est le nombre de modalités de la $K^{ième}$ variable ($J = \text{NUMVR}(K)$)

Le tableau est rangé de la manière suivante dans le fichier :

- Les NI premières lignes correspondent aux observations relatives aux NI individus de base.
- Les NICLA lignes suivantes correspondant aux NICLA individus supplémentaires.
- Les autres ne nous intéressent pas.

- Chaque ligne contient la suite d'entiers représentant les codages condensés des modalités prises par l'individu correspondant pour les variables de 1 à NUMVR(NV) au moins.

Le fichier des libellés.

Ce fichier n'est utilisé que si ILIB prend la valeur 1.

Son nom est donné par D.8

Il contient au moins NV libellés de variables et de l'information permettant de déterminer les nombres de modalités et les libellés des modalités de ces variables.

Si NELIB prend la valeur 1, il devra avoir la structure F1

Si NELIB prend la valeur 0, il devra avoir la structure F2.

Chaque nombre de modalités NMOD doit vérifier : 1 NMOD 9.

Structure F1.

- - - - -

ligne 1 : le nombre de modalités

ligne 2 : les libellés des modalités

(suite de groupes de 4 caractères séparés par un caractère ' '.

Le premier caractère de la ligne coïncide avec le premier caractère du premier libellé).

Pour I allant de 1 à NV :

ligne I+2 : libellé de la variable numéro I

(les 10 premiers caractères de la ligne)

Structure F2.

- - - - -

our I allant de 1 à NV :

ligne 2.I-1 : libellé de la variable n° I (les 10 premiers caractères de la ligne) et nombre de modalités pour cette variable (entier à partir du 12^{ième} caractère).

ligne 2.I : Les libellés des modalités de la variable n° I (même
forme que la ligne 2 de la structure Fl).

C.2 Pour exécuter le deuxième programme.

=====

Il suffit de taper la commande :

EXECUTE PRINC2.FOR, ANACOR.FOR, ANDISZ.FOR, SAVA.FOR, VALIDA.FOR, LECTUR.FOR,
MSG.FOR, SYS:EISDP/SEARCH

L'ordinateur enverra le message :

```

XXXXXXXXXXXXXXXXXXXXXXXXX
x METHODE DE SAPORTA. x
x PROGRAMME NUMERO 2. x
XXXXXXXXXXXXXXXXXXXXXXXXX

```

QUEL EST LE NOM DU FICHIER DE DONNEES ?

Il faudra répondre en donnant le nom qui avait été introduit en D.32

L'ordinateur s'occupe du reste.

take pcmd.CMD.1 (LOGGING OUTPUT ON)
LINK: Loading
ELNKXCT PRINC1 execution]

* METHODE DE SAFORTA. *
* PROGRAMME NUMERO 1. *

SI VOUS INTRODUISEZ LES DONNEES AU TERMINAL, TAPEZ 1, SINON TAPEZ 0
1

QUEL EST LE NOMBRE DE VARIABLES EXPLICATIVES A CONSIDERER ?
4

SI LES VARIABLES SE SUIVENT DANS LE FICHIER DES CODAGES CONDENSES,
TAPEZ 1, SINON TAPEZ 0
0

QUELS SONT LES NUMEROS DES VARIABLES A CONSIDERER ?
1,3,5,7

QUEL EST LE NOMBRE D'INDIVIDUS ?
32

QUEL EST LE NOMBRE D'INDIVIDUS SUPPLEMENTAIRES A CLASSER ?
0

QUEL EST LE NOM DU FICHIER DES POIDS DES INDIVIDUS ?
fpoids.dat

S'IL EXISTE UN FICHIER DE LIBELLES, TAPEZ 1, SINON TAPEZ 0.
0

SI TOUTES LES VARIABLES ONT LE MEME NOMBRE DE MODALITES,
TAPEZ 1, SINON TAPEZ 0
1

QUEL EST LE NOMBRE DE MODALITES ?
9

QUEL EST LE NOM DU FICHIER DES CODAGES CONDENSES ?
CODCON.DAT

QUEL EST LE NOMBRE DE GROUPES D'INDIVIDUS ?
2

QUEL EST LE NOMBRE D'INDIVIDUS DANS CHAQUE GROUPE ?
4,28

SI VOUS CHOISISSEZ LA METHODE D'ARRET PAR COEFFICIENT DE TSCHUPROW
MULTIPLE, TAPEZ 1. DANS LE CAS CONTRAIRE, TAPEZ 0.
0

QUEL EST LE NOMBRE DE VARIABLES A SELECTIONNER ?

2

QUEL EST LE POIDS MAXIMUM AUTORISE POUR LA
MODALITE NON-REPONDU D'UNE VARIABLE ?

.15

QUEL EST LE POIDS MINIMUM POUR QU'UNE MODALITE SOIT CONSERVEE ?

.01

SI VOUS CHOISISSEZ MAINTENANT LA METHODE POUR SUPPRIMER LES TROUS DANS
LE TABLEAU DES CODAGES CONDENSES,TAPEZ 1, SINON TAPEZ 0

1

QUELLE METHODE CHOISISSEZ-VOUS ?

SUPPRIMER L'INDIVIDU	----> 1
VENTILER AU HASARD	----> 2
REGROUPER DES MODALITES	----> 3

1

QUEL EST LE SEUIL A PARTIR DUQUEL ON CONSIDERE QUE DEUX
VARIABLES SONT REDONDANTES ?

.98

QUEL EST LE NOM DU FICHIER DES CODAGES CONDENSES APRES SELECTION ?
FCODN.DAT

QUEL EST LE NOM DU FICHIER DEVANT CONTENIR LE TABLEAU DE BURT
CROISANT LES MODALITES DES VARIABLES EXPLICATIVES
AVANT LA SELECTION ?
FBURTA.DAT

QUEL EST LE NOM DU FICHIER DEVANT CONTENIR LE TABLEAU DE BURT
CROISANT LES MODALITES DES VARIABLES EXPLICATIVES
APRES LA SELECTION ?
FBURT.DAT

QUEL EST LE NOM DU FICHIER DEVANT CONTENIR LE TABLEAU
DE BURT CROISANT LES MODALITES DES VARIABLES EXPLICATIVES
AVEC CELLES DE Y APRES LA SELECTION ?
FBURTY.DAT

QUEL EST LE NOM DU FICHIER DEVANT CONTENIR LES DONNEES
POUR LE SECOND PROGRAMME ?
FDON2.DAT

QUELLE EST L'INERTIE MINIMALE POUR UN FACTEUR Z ?
.00001

QUEL EST LE POUVOIR DISCRIMINANT CUMULE A ATTEINDRE
POUR LES FACTEURS Z ?
1.

QUEL EST LE MODE DE SORTIE POUR LES RESULTATS INTERMEDIAIRES ?
 TERMINAL --> 1
 IMPRIMANTE --> 2
 FICHIER --> 3

3

QUEL EST LE NOM DU FICHIER DES RESULTATS INTERMEDIAIRES ?

FRESUL.DAT

SI VOUS VOULEZ LA LISTE DES POIDS DES MODALITES APRES NETTOYAGE,
TAPEZ 1, SINON TAPEZ 0.
0

SI VOUS VOULEZ LE TABLEAU DES COEFFICIENTS DE TSCHUPROW DU PREMIER
ORDRE, TAPEZ 1, SINON TAPEZ 0
1

SI VOUS VOULEZ LES TABLEAUX DES COEFFICIENTS DE TSCHUPROW
PERIODIQUEMENT, DONNEZ LA PERIODE ≥ 0
0

SI VOUS VOULEZ LE TABLEAU DES COEFFICIENTS DE TSCHUPROW DU
PREMIER ORDRE AVEC Y, TAPEZ 1, SINON TAPEZ 0.
1

SI VOUS VOULEZ LES TABLEAUX DES COEFFICIENTS DE TSCHUPROW
AVEC Y PERIODIQUEMENT, TAPEZ 1, SINON TAPEZ 0.
1

SI VOUS VOULEZ LA VALEUR DU COEFFICIENT DE TSCHUPROW
MAXIMUM AVEC Y A CHAQUE ETAPE, TAPEZ 1, SINON TAPEZ 0.
1

SI VOUS VOULEZ LA VALEUR DU COEFFICIENT DE TSCHUPROW MULTIPLE
AVEC Y ET SA CROISSANCE A CHAQUE ETAPE, TAPEZ 1 SINON TAPEZ 0.
1

CPU time 4.32 Elapsed time 2:45.30

End of PCMD.CMD.1

@

ANNEXE D. SOURCES FORTRAN.

=====

On peut trouver successivement :

PRINC1.FOR

SAISIE.FOR

NETTOY.FOR

SELECT.FOR

SAVA.FOR

VALIDA.FOR

LECTUR.FOR

MSG.FOR

PRINC2.FOR

ANACOR.FOR

ANDISZ.FOR

Ces programmes sont écrits en FORTRAN-77. La place mémoire nécessaire à l'exécution du deuxième programme (qui demande le plus de place) est de l'ordre de 95K.

Si on désire transporter ce programme sur un autre ordinateur, il faudra modifier le contenu de ANACOR et de ANDISZ de façon à conserver les propriétés des arguments.


```

*****C
C*                                          *C
C*          METHODE DE SAPORTA, PREMIER PROGRAMME          *C
C*                                          *C
*****C

```

```

C      Ce programme comporte:
C          - la saisie des donnees
C          - le nettoyage des donnees
C          - la selection des variables explicatives.
C
C      Pour l'introduction des donnees, cfr. memoire annexe C.
C      Pour la structure du programme, cfr. memoire chapitre VII.
C      Pour le nettoyage et la selection des variables,
C          cfr. memoire chapitre III.
C
=====

```

```

PROGRAM PRTOC1

```

```

C      declarations.
C      -----

```

```

IMPLICIT DOUBLE PRECISION (A-H,O-Y)
IMPLICIT LOGICAL (Z)

```

```

COMMON/COM1/ITOT1,ITOT2,IMAR1,IMAR2,IODBT1,IODBT2,ICTP1
1,ICTP2,ICTR3,IPM1,IPM2,ICT11,ICT12,ICTY11,ICTY12
2,ICTMX1,ICTMX2,ICTMP1,ICTMP2,NOBT1,NOBT2,NOBT3

```

```

COMMON/COM2/ICT1,ICTY1,ICTMX,ICTMP,IMAR,ICTF,ICTVF,SP,OTM
1,NVS,VLPMTN,EDISCU

```

```

COMMON/COM3/ECODN,ETBTN,EBURT,EDON2,EBURTA,EBURTY

```

```

COMMON/COM6/NUMRES,NTUTM

```

```

COMMON/COM7/ECOD1

```

```

COMMON/COM4/PLRV,PMNTN,IODBT,ICTR,IPMN,MODGR,NUMTR

```

```

COMMON/COM8/ECOD,ETIP

```

```

COMMON/COM5/NV,NUMVR,NT,NICLA,PI,ILIP,MPDS,MGR,NVIGR

```

```

COMMON/COM9/ILIPVAR,LTSMOD

```

```

CHARACTER*(4) LIPMOD(900)
CHARACTER*(10) EDON,ECOD,LTBVAR(100),ECODN,ETIPN,EBURT,EDON2
1,ERES,ECOD1,ETIP,EBURTA,EBURTY

```

```

DIMENSION NUMVR(100),PI(4000),MPDS(101),NVIGR(9),MODGR(900)

```

```

DATA NUMBPT,NUMFD,NUMEP/3,51,52/
DATA NCXIN1,NCXIN2/0,1/

```

```

C      choix du mode de saisie
C      -----

```

```

NUMTRES

```

```

C      affichage du message d'entete
      CALL MSGT(NUMTER,1)
C      saisie du mode d'introduction ( ZP ne sert a rien ici ).
      CALL SAVAI(NUMTER,NUMTER,NUMTER,NCXIN,NCXIN1,NCXIN2,2,1,2,ZP)
C      affectation de NUMIN.
      IF (NCXIN.EQ.NCXIN1) THEN
          CALL SAVAS(NUMTER,NUMTER,NUMTER,FDON,3,3,4,ZP)
          OPEN(UNIT=NUMED,FILE=FDON,DEVICE='DSK')
          NUMIN=NUMED
      ELSE
          NUMIN=NUMTER
      END IF

C      saisie des donnees
C      -----

      ZTOTAL=.FALSE.
      CALL SAISIE(NUMIN,NGUT,FRES,ZTOTAL)
      IF (NUMIN.EQ.NUMED) CLOSE(NUMED)

C      arret en cas de probleme dans les donnees

      IF (ZTOTAL) THEN
          WRITE(NUMTER,*)
          WRITE(NUMTER,*) 'CORRIGEZ LES DONNEES ET RECOMMENCEZ !!'
          STOP
      END IF

      NGUTM=MPOS(NGV+1)

C      determination du mode de sortie

      IF (NGUT.EQ.NGUT3) THEN
          OPEN (UNIT=NUMER,FILE=FRES,DEVICE='DSK')
          NUMRES=NUMER
      ELSE
          IF (NGUT.EQ.NGUT1) THEN
              NUMRES=NUMTER
          ELSE
              NUMRES=NUMIPT
          END IF
      END IF

C      nettoyage des donnees
C      -----

      CALL NETTOY(ZTOTAL)

C      arret en cas de probleme lors du nettoyage

      IF (ZTOTAL) THEN
          WRITE(NUMTER,*)
          WRITE(NUMTER,*) 'CORRIGEZ LES DONNEES ET RECOMMENCEZ !!'
          IF (NUMRES.EQ.NUMER) CLOSE(NUMRES)
          STOP
      END IF

C      arret si le nombre de groupes d'individus est inferieur a 2

      IF (NGR.LE.1) THEN

```



```

WRITE(NUMTER,*)
WRITE(NUMTER,*)'ANALYSE DISCRIMINANTE INUTILE VU LE NOMBRE
1 DE GROUPES QUI RESTE !!'
IF (NUMRES.EQ.NUMER) CLOSE(NUMRES)
STOP
END IF

```

C arret s'il ne reste plus de variable.

```

IF (NV.EQ.0) THEN
WRITE(NUMTER,*)
WRITE(NUMTER,*)'NETTOYAGE TROP VIGOUPEUX, IL NE RESTE
1 PLUS RIEN !!'
IF (NUMRES.EQ.NUMER) CLOSE(NUMRES)
STOP
END IF

```

C selection des variables discriminantes.

C -----

```

CALL SELECT(NOUT,RES)

```

C IF (NUMRES.EQ.NUMER) CLOSE(NUMRES)

```

END

```

C affectation des constantes au COMMON 1.

C -----

```

BLOCK DATA
COMMON/COM1/ILIB1,ILIB2,IMAR1,IMAR2,IQDBT1,IQDBT2,ICHTR1
1,ICHTR2,ICHTR3,IPMN1,IPMN2,ICT11,ICT12,ICTY11,ICTY12
2,ICTMX1,ICTMX2,ICTMP1,ICTMP2,NOUT1,NOUT2,NOUT3
DATA ILIB1,ILIB2,IMAR1,IMAR2,IQDBT1,IQDBT2,ICHTR1
1,ICHTR2,ICHTR3,IPMN1,IPMN2,ICT11,ICT12,ICTY11,ICTY12
2,ICTMX1,ICTMX2,ICTMP1,ICTMP2,NOUT1,NOUT2,NOUT3
3/0,1,0,1,0,1,1,2,3,0,1,0,1,0,1,0,1,0,1,1,2,3/
END

```

```

C*****C
C*                                           *C
C*           SAISIE DES DONNEES.           *C
C*                                           *C
C*****C

```

SUBROUTINE SAISIE(NUMIN,NOUT,FRES,ZTOTAL)

```

C   Declarations.
C   -----

```

```

IMPLICIT DOUBLE PRECISION (A-H,O-Y)
IMPLICIT LOGICAL (Z)

```

```

COMMON/COM1/ILIB1,ILIB2,IMAR1,IMAR2,IQDBT1,IQDBT2,ICHTR1
1,ICHTR2,ICHTR3,IPMN1,IPMN2,ICT11,ICT12,ICTY11,ICTY12
2,ICTMX1,ICTMX2,ICTMP1,ICTMP2,NOUT1,NOUT2,NOUT3

```

```

COMMON/COM2/ICT1,ICTY1,ICTMX,ICTMP,IMAR
1,ICTF,ICTYF,SR,QTM,NVS,VLPMIN,PDISCU

```

```

COMMON/COM3/FCODN,FLIBN,FBURT,FDON2,FBURTA,FBURTY

```

```

COMMON/COM4/PNRV,FMMIN,IQDBT,ICHTR,IPMN,MODGR,NUMTER

```

```

COMMON/COM5/NU,NUMVR,NI,NICLA,PI,ILIB,MPOS,NGR,NVIGR

```

```

COMMON/COM8/FCOD,FLIB

```

```

COMMON/COM9/LIBVAR,LIBMOD

```

```

CHARACTER*(10) FPDS,LIBVAR(100),FCOD,FLIB
CHARACTER*(4) LAUMOD(9),LIBMOD(900)
CHARACTER*(*) FRES
CHARACTER*(10) FCODN,FLIBN,FBURT,FDON2,FBURTA,FBURTY

```

```

DIMENSION NUMVR(100),PI(4000),MPOS(101),NUMOD(100)
1,NCODI(100),NVIGR(9),NREGR(9),MODGR(900)

```

```

DATA NUMFP,NUMFL,NUMCC/53,54,55/
DATA NELIB1,NELIB2/0,1/
DATA NUMIN,NUMAX,NVSMIN/1,100,1/
DATA NIMIN,NIMAX/2,4000/
DATA NICLMI/0/
DATA MODMIN,MODMAX/1,9/
DATA NGRMIN,NGRMAX/2,9/
DATA PNRMIN,PNRMAX/0.D0,1.D0/
DATA PMMINF,PMMSUP,EPSPMM/0.D0,1.D0,1.D-15/
DATA ITPGR1,ITPGR2/0,1/
DATA MVAR1,MVAR2/0,1/
DATA NEMOD1,NEMOD2/0,1/
DATA PIINF,PISUP,EPSPPI,SOMPI,EPSSPI/0.D0,1.D0,1.D-15
1,1.D0,1.D-15/
DATA QTMINF,QTMSUP,EPSSQM/0.D0,1.D0,1.D-15/
DATA SRINF,SRSUP,EPSSR/.5D0,1.D0,1.D-15/
DATA ICTFMN,ICTYFM/0,0/
DATA VLPINF,VLPSUP,EPVLP/0.D0,1.D0,1.D-15/
DATA PDCINF,EPSPDC/0.D0,1.D-15/

```

```

C   initialisation de l'indicateur d'erreur principal.

```



```

ZTOTAL=.FALSE.

C   saisie et validation du nombre de variables a considerer, NV.
C   -----

CALL SAVA2(NUMTER,NUMIN,NUMTER,NV,NUMIN,NUMAX,4,5,6,ZNV)

ZTOTAL=ZTOTAL.OR.ZNV

C   saisie et validation du mode de repartition des variables
C   dans le fichier, MODVAR.
C   -----

CALL SAVA1(NUMTER,NUMIN,NUMTER,MODVAR,MVAR1,MVAR2
1,5,7,8,ZMVAR)

ZTOTAL=ZTOTAL.OR.ZMVAR

C   si on ne dispose pas de MODVAR, il est impossible de
C   continuer la saisie.

IF (ZMVAR) THEN
    CALL MSGERR(NUMTER,10)
    RETURN
END IF

C   affectation du vecteur des numeros des variables.
C   -----

C   si NV n'est pas disponible, il est impossible de saisir les
C   numeros des variables.

IF (ZNV) THEN
    CALL MSGERR(NUMTER,9)
    READ(NUMIN,*,ERR=999,END=999)
    ZNUMVR=.TRUE.
ELSE

C   cas ou les variables ne se suivent pas dans le fichier,
C   saisie des numeros des variables.

    IF (MODVAR.EQ.MVAR1) THEN
        CALL SAVA13(NUMTER,NUMIN,NUMTER,NUMVR,NV,NUMAX,NUMIN
1 ,NUMAX,6,11,12,13,ZNUMVR)
    ELSE

C   cas ou les variables se suivent dans le fichier.

        ZNUMVR=.FALSE.
        DO 5 I=1,NV
            NUMVR(I)=I
5        CONTINUE
        END IF

    END IF

ZTOTAL=ZTOTAL.OR.ZNUMVR

C   saisie et validation du nombre d'individus, NI.
C   -----

```

```

CALL SAVA2(NUMTER,NUMIN,NUMTER,NI,NIMIN,NIMAX,7,14,15,ZNI)

ZTOTAL=ZTOTAL.OR.ZNI

C      saisie et validation du nombre d'individus supplementaires
C      a classer, NICLA.
C      -----

CALL SAVA4(NUMTER,NUMIN,NUMTER,NICLA,NICLMI,40,95,96,ZNICLA)

ZTOTAL=ZTOTAL.OR.ZNICLA

C      saisie et validation du nom du fichier des poids des
C      individus, FPDS.
C      -----

CALL SAVA5(NUMTER,NUMIN,NUMTER,FPDS,8,16,17,ZFPDS)

ZTOTAL=ZTOTAL.OR.ZFPDS

C      affectation du vecteur des poids des individus.
C      -----

C      si NI ou FPDS ne sont pas disponibles, l'affectation ne peut
C      etre effectuee

      IF (ZNI.OR.ZFPDS) THEN
        ZPI=.TRUE.
        CALL MSGERR(NUMTER,97)
      ELSE

C      saisie et validation du vecteur des poids des individus.

        OPEN(UNIT=NUMFP,FILE=FPDS,DEVICE='DSK')
        CALL SAVA14(NUMTER,NUMFP,NUMTER,PI,NI,NIMAX,FIINF,PISUP
1,EPSP,ISOMPI,EPSSPI,0,98,99,100,ZPI)
        CLOSE(NUMFP)
      END IF

      ZTOTAL=ZTOTAL.OR.ZPI

C      saisie et validation de l'indicateur de l'existence d'un
C      fichier de libelles.
C      -----

CALL SAVA1(NUMTER,NUMIN,NUMTER,ILIB,ILIB1,ILIB2,9,18,19
1,ZILIB)

ZTOTAL=ZTOTAL.OR.ZILIB

C      si ILIB n'est pas disponible, il est impossible de continuer
C      la saisie.

      IF (ZILIB) THEN
        CALL MSGERR(NUMTER,10)
        RETURN
      END IF

C      cas ou il y a utilisation d'un fichier de libelles.
C      =====

```



```

IF (ILIB.EQ.ILIB2) THEN
C
C   saisie et validation du nom du fichier des libelles.
C   -----
      CALL SAVAS(NUMTER,NUMIN,NUMTER,FLIB,10,20,21,ZFLIB)
C
C   saisie et validation de l'indicateur de l'escalite des
C   libelles de modalites
C   -----
      CALL SAVA1(NUMTER,NUMIN,NUMTER,NELIB,NELIB1,NELIB2
1 ,11,22,23,ZNELIB)
C
C   saisie et validation du contenu du fichier des libelles.
C   -----
C
C   si flib, nelib, nv ou numvr sont indisponibles, il est
C   impossible de saisir et de valider le contenu de ce fichier
      IF (ZFLIB.OR.ZNELIB.OR.ZNV.OR.ZNUMVR) THEN
        CALL MSGERR(NUMTER,101)
        ZCLIB=.TRUE.
      ELSE
C
C   cas ou la saisie est possible
        OPEN(UNIT=NUMFL,FILE=FLIB,DEVICE='DSK')
        IF (NELIB.EQ.NELIB2) THEN
C
C   cas ou les libelles de modalites sont les memes pour
C   toutes les variables.
C   -----
          ZCLIB=.FALSE.
C
C   saisie et validation du nombre de modalites.
          CALL SAVA2(NUMTER,NUMFL,NUMTER,NBMOD
1 ,MODMIN,MODMAX,0,103,104,ZCLIB)
C
C   si la lecture du nombre de modalites s'est effectuee
C   correctement, saisie et validation des libelles des
C   modalites.
          IF (.NOT.ZCLIB) THEN
            CALL LEC8(NUMFL,LIBMOD,NBMOD,MODMAX*NUMAX,ZCLIB)
            IF (ZCLIB) CALL MSGERR(NUMTER,105)
          END IF
C
C   lecture et validation des libelles des variables s'il
C   n'y a pas eu d'erreur.
          I=1
          I1=1
          MPOS(1)=0
          DO 10 WHILE (.NOT.ZCLIB.AND.(I1.LE.NV))
            IF (NUMVR(I1).EQ.I) THEN
              CALL LEC6(NUMFL,LIBVAR(I1),ZCLIB)

```

```

        I1=I1+1
    ELSE
        READ(NUMFL,*,ERR=998,END=998)
    END IF
    IF (ZCLIB) CALL MSGERR(NUMTER,102)
    I=I+1
10    CONTINUE

C      si le contenu du fichier des libelles est valide,
C      affectation:
C          - du vecteur des nombres de modalites
C          - du vecteur des positions des modalites
C          - du vecteur des libelles des modalites

    IF (,NOT,ZCLIB) THEN
        DO 20 I=1,NV
            NVMOD(I)=NBMOD
            MPOS(I+1)=MPOS(I)+NBMOD
            DO 30 J=1,NBMOD
                LIBMOD((I-1)*NBMOD+J)=LIBMOD(J)
30            CONTINUE
20        CONTINUE
    END IF

C      ces ou les libelles des modalites ne sont pas les
C      memes pour toutes les variables.
C      -----

ELSE
    ZCLIB=.FALSE.

C      saisie et validation du contenu du fichier

    I=1
    I1=1
    MPOS(1)=0

C      pour chaque variable:

    DO 40 WHILE (,NOT,ZCLIB.AND.(I1,LE,NV))
        IF (NUMVR(I1).EQ,I) THEN

C          lecture et validation de son libelle et du nombre de
C          modalites

            CALL LEC7(NUMFL,LIBVAR(I1),NVMOD(I1),ZCLIB)
            IF (ZCLIB) THEN
                CALL MSGERR(NUMTER,106)
            ELSE

C          si le libelle et le nombre de modalites sont
C          corrects, mise a jour du vecteur des positions des
C          modalites et saisie des libelles des modalites

                MPOS(I1+1)=MPOS(I1)+NVMOD(I1)
                CALL LEC8(NUMFL,LAUMOD,NVMOD(I1),MODMAX,ZCLIB)

C          s'il y a eu un probleme lors de la lecture des
C          libelles,                                -> message

```



```

      IF (ZCLIB) THEN
        CALL MSGERR(NUMTER,105)

C      sinon affectation du vecteur des libelles des
C      modalites et passage a la variable suivante.

        ELSE
          DO 50 J=1,NVMOD(I1)
            LIBMOD(MPOS(I1)+J)=LAUMOD(J)
50          CONTINUE
          END IF
        END IF
        I1=I1+1
      ELSE
        READ(NUMFL,*,ERR=998,END=998)
        READ(NUMFL,*,ERR=998,END=998)
        END IF
        I=I+1
40        CONTINUE
      END IF
      GOTO 997
998      ZCLIB=.TRUE.
      CALL MSGERR(NUMTER,121)
997      CLOSE(NUMFL)
      END IF
      ZNEMOD=.FALSE.
      ZLIB=.FALSE.
      ZNVMOD=.FALSE.

C      cas ou il n'y a pas utilisation d'un fichier de libelles.
C      =====

      ELSE
        ZCLIB=.FALSE.
        ZNELIB=.FALSE.
        ZFLIB=.FALSE.

C      saisie et validation de l'indicateur de l'egalite des
C      nombres de modalites.
C      -----

        CALL SAVA1(NUMTER,NUMIN,NUMTER,NEMOD,NEMOD1,NEMOD2
1      ,12,24,25,ZNEMOD)

C      si NEMOD est indisponible, il est impossible de valider les
C      nombres de modalites

        IF (ZNEMOD) THEN
          CALL MSGERR(NUMTER,26)
          READ(NUMIN,*,ERR=999,END=999)
          ZNBMOD=.TRUE.
          ZNVMOD=.TRUE.
        ELSE

C      saisie et validation du nombre de modalites.
C      -----

          IF (NEMOD.EQ.NEMOD2) THEN
            ZNVMOD=.FALSE.
            CALL SAVA2(NUMTER,NUMIN,NUMTER,NBMOD,MODMIN,MODMAX
1            ,13,27,28,ZNBMOD)

```

```

IF (ZNV,OR,ZNVVR,OR,ZNVVVR,OR,ZNI,OR,ZNICL,OR,ZFLIB
1,OR,ZNELIB,OR,ZCLIB,OR,ZNEMOD,OR,ZNBMOD,OR,ZNVMOD) THEN
CALL M3ERRR(NUMTER,107)
ZCODBC=.TRUE.

```

```

CALL SAVAS(NUMTER,NUMIN,NUMTER,FCOD,15,32,33,ZFCOD)

```

```

-----
saisie et validation du nom du fichier des codes condensés,
FCOD,

```

```

ZTOTAL=ZTOTAL,OR,ZFLIB,OR,ZNELIB,OR,ZCLIB,OR,ZNEMOD
1,OR,ZNBMOD,OR,ZNVMOD

```

```

END IF
END IF
END IF
END IF
ZNBMOD=.FALSE.

```

```

END IF
CONTINUE
MPOS(I+1)=MPOS(I)+NVMOD(I)
DO 42 I=1,NV
IF ((.NOT.ZNV).AND.(.NOT.ZNVMOD)) THEN

```

```

vecteur des positions des modèles,
si NV et NVMOD sont disponibles, affectation du

```

```

CALL SAVAS(NUMTER,NUMIN,NUMTER,NV,NVMAX,KODMIN
,NVMAX,14,29,30,ZNVMOD)

```

```

cas où la saisie est possible,

```

```

ELSE
ZNVMOD=.TRUE.
ZNBMOD=.TRUE.
READ(NUMIN,*,ERR=999,END=999)
CALL M3ERRR(NUMTER,31)
IF (ZNV) THEN

```

```

si NV n'est pas disponible, cette saisie est impossible,

```

```

ELSE

```

```

-----
saisie et validation des nombres de modèles,

```

```

END IF
CONTINUE
NVMOD(I)=NBMOD
MPOS(I+1)=MPOS(I)+NBMOD
DO 41 I=1,NV
IF ((.NOT.ZNV).AND.(.NOT.ZNBMOD)) THEN

```

```

positions des modèles,
vecteur des nombres de modèles et du vecteur des
si NV et NBMOD sont disponibles, affectation du

```



```

C      validation du contenu du fichier des codages condenses.
C      -----

ELSE
  OPEN(UNIT=NUMCC,FILE=FCOD,DEVICE='DSK')
  ZCODC=,FALSE.

C      pour chaque ligne, tant qu'il n'y a pas d'erreur:

  I=1
  DO 60 WHILE ((.NOT.ZCODC).AND.(I.LE.(NI+NICLA)))

C      saisie de la ligne.

      CALL LEC2(NUMCC,NCODI,NUMVR(NV),NUMAX,ZCODC)

      IF (ZCODC) THEN
        CALL MSGERR(NUMTER,108)
      ELSE

C      si la saisie s'est effectuee correctement, validation de
C      chaque element de la ligne tant qu'il n'y a pas d'erreur.

        J=1
        DO 70 WHILE ((.NOT.ZCODC).AND.(J.LE.NV))
          IF ((NCODI(NUMVR(J)).LT.MODMIN).OR.(NCODI(NUMVR(J))
1          .GT.NVMOD(J))) THEN
            ZCODC=,TRUE.
            CALL MSGERR(NUMTER,109)
          ELSE
            J=J+1
          END IF
70        CONTINUE
        END IF
        I=I+1
60      CONTINUE

      CLOSE(NUMCC)

  END IF

  ZTOTAL=ZTOTAL.OR.ZFCOD.OR.ZCODC

C      saisie et validation du nombre de groupes d'individus, NGR.
C      -----

  CALL SAVA2(NUMTER,NUMIN,NUMTER,NGR,NGRMIN,NGRMAX
1,16,34,35,ZGR)

  ZTOTAL=ZTOTAL.OR.ZGR

C      saisie et validation de la liste des nombres d'individus dans
C      chaque groupe
C      -----

C      si GR ou NI sont indisponibles, cette saisie est impossible

  IF (ZGR.OR.ZNI) THEN
    CALL MSGERR(NUMTER,36)
    READ(NUMIN,*,ERR=999,END=999)
  
```

ZNVIGR=.TRUE.

C cas ou la saisie est possible.

ELSE

CALL SAVA10(NUMTER,NUMIN,NUMTER,NVIGR,NGR,NGRMAX,NIMIN-1
1 ,NI-1,NI,17,37,38,39,ZNVIGR)
END IF

ZTOTAL=ZTOTAL.OR.ZNVIGR

C saisie et validation du choix de la methode d'arret.
C -----

CALL SAVA1(NUMTER,NUMIN,NUMTER,IMAR,IMAR1,IMAR2
1,18,40,41,ZIMAR)

ZTOTAL=ZTOTAL.OR.ZIMAR

C si IMAR est indisponible, il est impossible de saisir et de
C valider QTM ou NVS

IF (ZIMAR) THEN
CALL MSGERR(NUMTER,42)
READ(NUMIN,*,ERR=999,END=999)
ZQTM=.TRUE.
ZNVS=.TRUE.
ELSE

C saisie et validation de la quantite fixant la croissance
C minimale pour le coefficient de tschuprow multiple, QTM.
C -----

IF (IMAR.EQ.IMAR2) THEN
CALL SAVA6(NUMTER,NUMIN,NUMTER,QTM,QTMINF,QTMSUP,EPSQTM
1 ,20,46,47,ZQTM)
ZNVS=.FALSE.

ELSE

C saisie et validation du nombre de variables a selectionner,
C NVS.
C -----

C si NV est indisponible, la validation de NVS est impossible

IF (ZNV) THEN
CALL MSGERR(NUMTER,43)
READ(NUMIN,*,ERR=999,END=999)
ZNVS=.TRUE.
ZQTM=.TRUE.
ELSE

C cas ou la validation est possible

CALL SAVA2(NUMTER,NUMIN,NUMTER,NVS,NVSMIN,NV,19,44,45
1 ,ZNVS)
ZQTM=.FALSE.
END IF
END IF
END IF

ZTOTAL=ZTOTAL.OR.ZNVS.OR.ZQTM

C saisie et validation du poids maximum pour la modalite non-
C repondu d'une variable, PNRV.
C -----

CALL SAVA7(NUMTER,NUMIN,NUMTER,PNRV,PNRMIN,PNRMAX
1,21,48,49,ZPNRV)

ZTOTAL=ZTOTAL.OR.ZPNRV

C saisie et validation du poids minimum pour qu'une modalite
C soit conservee, PMMIN.
C -----

CALL SAVA6(NUMTER,NUMIN,NUMTER,PMMIN,PMMINF,PMMSUP,EPSPMM
1,22,50,51,ZPMMIN)

ZTOTAL=ZTOTAL.OR.ZPMMIN

C saisie et validation de l'indicateur de "quand" on choisit la
C methode pour boucher les trous dans le tableau des codases
C condenses ?, IQDBT.
C -----

CALL SAVA1(NUMTER,NUMIN,NUMTER,IQDBT,IQDBT1,IQDBT2
1,23,52,53,ZIQDBT)

ZTOTAL=ZTOTAL.OR.ZIQDBT

C si IQDBT n'est pas disponible, il est impossible de continuer
C la saisie.

IF (ZIQDBT) THEN
CALL MSGERR(NUMTER,10)
RETURN
END IF

C cas ou on choisit la methode avant l'execution.
C =====

ICHTR=0
IF (IQDBT.EQ.IQDBT2) THEN

C saisie et validation du choix de la methode.
C -----

CALL SAVA3(NUMTER,NUMIN,NUMTER,ICHTR,ICHTR1,ICHTR2,ICHTR3
1,24,54,55,ZICHTR)

ZTOTAL=ZTOTAL.OR.ZICHTR

C si ICHTR est indisponible, il est impossible de continuer
C la saisie.

IF (ZICHTR) THEN
CALL MSGERR(NUMTER,10)
RETURN
END IF


```

C      cas ou on a choisi la troisieme methode.
C
C      saisie et validation du type de regroupement.
      IF (ICHTR.EQ.ICHTR3) THEN
        CALL SAVA1(NUMTER,NUMIN,NUMTER,ITPGR,ITPGR1,ITPGR2,48,110
1      ,111,ZITPGR)

        ZTOTAL=ZTOTAL.OR.ZITPGR

C      si ITPGR est indisponible, il est impossible de continuer
C      la saisie

        IF (ZITPGR) THEN
          CALL MSGERR(NUMTER,10)
          RETURN
        END IF

C      cas ou la saisie et la validation des modalites groupes
C      est impossible et ou eventuellement on est oblige
C      d'arreter la saisie.

1      IF (ZNV.OR.ZFLIB.OR.ZNELIB.OR.ZCLIB.OR.ZNEMOD
        .OR.ZNBMOD.OR.ZNVMOD) THEN
        CALL MSGERR(NUMTER,112)

        IF (ZNV) THEN
          CALL MSGERR(NUMTER,10)
          RETURN
        END IF

        IF (ITPGR.EQ.ITPGR2) THEN
          READ(NUMIN,*)
        ELSE
          DO 75 I=1,NV
            READ(NUMIN,*)
75          CONTINUE
          END IF

          ZNREGR=.TRUE.
        ELSE

C      saisie et validation des modalites groupes (les memes
C      pour chaque variables).
C      -----

1      IF (ITPGR.EQ.ITPGR2) THEN
        CALL SAVA11(NUMTER,NUMIN,NUMTER,NREGR,NVMOD(1),MODMAX
        ,MODMIN,NVMOD(1),49,113,114,115,ZNREGR)

C      si on n'a pas declare que les variables avaient meme
C      nombre de modalites, les modalites groupes ne peuvent
C      pas etre les memes.

1      IF (((ILIB.EQ.ILIB2).AND.(.NOT.ZNELIB).AND.(NELIB.EQ.
2      NELIB2)).OR.((ILIB.EQ.ILIB1).AND.(.NOT.ZNEMOD).AND.
        (NEMOD.EQ.NEMOD1))) THEN
        CALL MSGERR(NUMTER,122)
        ZNREGR=.TRUE.
      END IF

```



```

C      si NREGR est disponible, affectation du vecteur des
C      modalites groupes des modalites des differentes
C      variables.

      IF (.NOT.ZNREGR) THEN
        DO 77 I=1,NV
          DO 77 J=1,NVMOD(1)
            MODGR((I-1)*NVMOD(1)+J)=NREGR(J)
77      CONTINUE
        END IF

      ELSE

C      saisie et validation des modalites groupes pour chaque
C      variable.

      I=1
      ZNREGR=.FALSE.

C      pour chaque variable, tant qu'il n'y a pas d'erreur:

      DO 80 WHILE (.NOT.ZNREGR.AND.(I.LE.NV))

C      lecture des modalites groupes de ces modalites

          IF (NUMIN.EQ.NUMTER) WRITE(NUMTER,*)'MODALITES
1 GROUPES POUR LA VARIABLE ',I
          CALL SAVA11(NUMTER,NUMIN,NUMTER,NREGR,NVMOD(I)
1          ,MODMAX,MODMIN,NVMOD(I),0,113,114,115,ZNREGR)

C      si les modalites groupes des modalites de cette
C      variable sont disponibles on les insere dans le
C      vecteur des modalites groupes.

          IF (.NOT.ZNEGR) THEN
            DO 90 J=1,NVMOD(I)
              MODGR(MPOS(I)+J)=NREGR(J)
90          CONTINUE
            END IF
            I=I+1
80          CONTINUE
          END IF
        END IF

      ELSE
        ZNREGR=.FALSE.
        ZITPGR=.FALSE.
      END IF
    ELSE
      ZICHTR=.FALSE.
      ZITPGR=.FALSE.
      ZNREGR=.FALSE.
    END IF

    ZTOTAL=ZTOTAL.OR.ZITPGR.OR.ZNREGR

C      saisie et validation du seuil de redondance, SR.
C      -----

    CALL SAVA6(NUMTER,NUMIN,NUMTER,SR,SRINF,SRSUP,EPSSR
1,25,56,57,ZSR)

```


ZTOTAL=ZTOTAL.OR.ZSR

C saisie du nom du fichier des codases condenses apres
C selection, FCODN.
C -----

CALL SAVA12(NUMTER,NUMIN,NUMTER,FCODN,26,58,ZFCODN)

ZTOTAL=ZTOTAL.OR.ZFCODN

C saisie du nom du fichier des libelles apres selection, FLIBN.
C -----

IF (ILIB.EQ.ILIB2) THEN

CALL SAVA12(NUMTER,NUMIN,NUMTER,FLIBN,27,59,ZFLIBN)

ZTOTAL=ZTOTAL.OR.ZFLIBN

END IF

C saisie du nom du fichier devant contenir le tableau de burt
C croisant les modalites des variables explicatives avant la
C selection, FBURTA.
C -----

CALL SAVA12(NUMTER,NUMIN,NUMTER,FBURTA,52,123,ZBURTA)

ZTOTAL=ZTOTAL.OR.ZBURTA

C saisie du nom du fichier devant contenir le tableau de burt
C croisant les modalites des variables explicatives apres la
C selection, FBURT.
C -----

CALL SAVA12(NUMTER,NUMIN,NUMTER,FBURT,28,60,ZFBURT)

ZTOTAL=ZTOTAL.OR.ZFBURT

C saisie du nom du fichier devant contenir le tableau de burt
C croisant les modalites des variables explicatives avec celles
C de y apres la selection.
C -----

CALL SAVA12(NUMTER,NUMIN,NUMTER,FBURTY,32,65,ZBURTY)

ZTOTAL=ZTOTAL.OR.ZBURTY

C saisie du nom du fichier des donnees pour le second program-
C me, FDON2.
C -----

CALL SAVA12(NUMTER,NUMIN,NUMTER,FDON2,29,61,ZFDON2)

ZTOTAL=ZTOTAL.OR.ZFDON2

C saisie de l'inertie minimale pour qu'un facteur z soit
C conserve.
C -----

CALL SAVA6(NUMTER,NUMIN,NUMTER,VLPMIN,VLPINF,VLPSUP
1,EPSVLP,36,73,74,ZVLP)

ZTOTAL=ZTOTAL.OR.ZVLP

C saisie du pouvoir discriminant cumule a atteindre pour les
C facteurs z
C -----

CALL SAVA6(NUMTER,NUMIN,NUMTER,PDISCU,PDCINF,DFLOAT(NGR-1)
1+1.D-8,EPSPDC,37,75,76,ZPDC)

ZTOTAL=ZTOTAL.OR.ZPDC

C saisie et validation du choix du mode de sortie des resultats
C intermediaires, NOUT.
C -----

CALL SAVA3(NUMTER,NUMIN,NUMTER,NOUT,NOUT1,NOUT2,NOUT3
1,30,62,63,ZNOUT)

ZTOTAL=ZTOTAL.OR.ZNOUT

C si NOUT est indisponible, il est impossible de continuer la
C saisie.

IF (ZNOUT) THEN
CALL MSGERR(NUMTER,10)
RETURN
END IF

C saisie du nom du fichier des resultats intermediaires, FRES.
C -----

IF (NOUT.EQ.NOUT3) THEN
CALL SAVA12(NUMTER,NUMIN,NUMTER,FRES,31,64,ZFRES)

ZTOTAL=ZTOTAL.OR.ZFRES
END IF

C saisie et validation d'informations complementaires.
C -----

C on veut ou non la liste des poids des modalites apres
C nettoyage.

CALL SAVA1(NUMTER,NUMIN,NUMTER,IPMN,IPMN1,IPMN2
1,35,71,72,ZIPMN)
ZTOTAL=ZTOTAL.OR.ZIPMN

C on veut ou non le tableau des coefficients de tschuprow du
C premier ordre entre les variables

CALL SAVA1(NUMTER,NUMIN,NUMTER,ICT1,ICT11,ICT12
1,39,79,80,ZICT1)
ZTOTAL=ZTOTAL.OR.ZICT1

C periode avec laquelle on desire avoir le tableau des coeffi-
C cients de tschuprow entre les variables

CALL SAVA4(NUMTER,NUMIN,NUMTER,ICTF,ICTFMN
1,41,83,84,ZICTF)
ZTOTAL=ZTOTAL.OR.ZICTF

C on veut ou non le vecteur des coefficients de tschuprow avec
C y.

```
CALL SAVA1(NUMTER,NUMIN,NUMTER,ICTY1,ICTY11,ICTY12  
1,42,85,86,ZICTY1)  
ZTOTAL=ZTOTAL.OR.ZICTY1
```

C periode avec laquelle on desire le vecteur des coefficients
C de tschuprow avec y.

```
CALL SAVA4(NUMTER,NUMIN,NUMTER,ICTYF,ICTYFM  
1,43,87,88,ZICTYF)  
ZTOTAL=ZTOTAL.OR.ZICTYF
```

C on veut ou non la valeur du coefficient du tschuprow avec y
C maximum a chaque etape.

```
CALL SAVA1(NUMTER,NUMIN,NUMTER,ICTMX,ICTMX1,ICTMX2  
1,44,89,90,ZICTMX)  
ZTOTAL=ZTOTAL.OR.ZICTMX
```

C on veut ou non la valeur du coefficient de tschuprow multiple
C et sa croissance a chaque etape.

```
CALL SAVA1(NUMTER,NUMIN,NUMTER,ICTMP,ICTMP1,ICTMP2  
1,46,93,94,ZICTMP)  
ZTOTAL=ZTOTAL.OR.ZICTMP
```

RETURN

```
999 ZTOTAL=.TRUE.  
CALL MSGERR(NUMTER,120)  
RETURN  
END
```

C+++++
@


```

C*****C
C*                                           *C
C*           NETTOYAGE DES DONNEES.         *C
C*                                           *C
C*****C

```

```

SUBROUTINE NETTOY(ZTOTAL)

```

```

C      declarations
C      -----

```

```

IMPLICIT DOUBLE PRECISION (A-H,O-Y)
IMPLICIT LOGICAL (Z)

```

```

COMMON/COM1/ILIB1,ILIB2,IMAR1,IMAR2,IQDBT1,IQDBT2,ICHTR1
1,ICHTR2,ICHTR3,IPMN1,IPMN2

```

```

COMMON/COM4/PNRV,PMMIN,IQDBT,ICHTR,IPMN,MODGR,NUMTER

```

```

COMMON/COM5/NV,NUMVR,NI,NICLA,PI,ILIB,MPOS,NGR,NVIGR

```

```

COMMON/COM6/NUMRES,NTOTM

```

```

COMMON/COM7/FCOD1

```

```

COMMON/COM8/FCOD,FLIB

```

```

COMMON/COM9/LIBVAR,LIBMOD

```

```

CHARACTER*(10) LIBVAR(100),FCOD1,FCOD,FLIB,FCOD2
CHARACTER*(4) LIBMOD(900)

```

```

DIMENSION MPOS(101),PM(900),NICON(900),NCODI(100),NUMVR(100)
1,PI(4000),MPOSVG(101),MODGR(900),MPMG(900),MPVG(100)
2,NVIGR(9)

```

```

DATA NUMCC,NUMCC1,NUMCC2/56,57,58/

```

```

WRITE(NUMRES,*)'NOMBRE DE VARIABLES : ',NV
WRITE(NUMRES,*)'NUMEROS DES VARIABLES : '
WRITE(NUMRES,*)' ',(NUMVR(I),I=1,NV)
WRITE(NUMRES,*)'NOMBRE D''INDIVIDUS : ',NI
WRITE(NUMRES,*)'NOMBRE D''INDIVIDUS SUPPLEMENTAIRES : ',NICLA
WRITE(NUMRES,*)'POIDS DES INDIVIDUS : '
WRITE(NUMRES,*)(PI(I),I=1,NI)
IF (ILIB.EQ.ILIB2) THEN
  WRITE(NUMRES,*)'NOM DU FICHIER DES LIBELLES AVANT NET
1TOYAGE: ',FLIB
END IF
WRITE(NUMRES,*)'NOMBRES DE MODALITES POUR CHAQUE VARIABLE : '
DO 287 I=1,NV
WRITE(NUMRES,*)'  VARIABLE ',I,' --> ',MPOS(I+1)-MPOS(I),
1' MODALITES'
CONTINUE
WRITE(NUMRES,*)'NOM DU FICHIER DES CODAGES CONDENSES AVANT
1 NETTOYAGE : ',FCOD
WRITE(NUMRES,*)'NOMBRE DE GROUPES D''INDIVIDUS : ',NGR
WRITE(NUMRES,*)'NOMBRE D''INDIVIDUS DANS CHAQUE GROUPE : '
DO 286 I=1,NGR
WRITE(NUMRES,*)'  GROUPE ',I,' --> ',NVIGR(I)

```


CONTINUE

```
WRITE(NUMRES,*)'POIDS MAXIMUM AUTORISE POUR LA MODALITE
1 ''NON-REPONDU'' : ',PNRV
WRITE(NUMRES,*)'POIDS MINIMUM POUR QU''UNE MODALITE SOIT
1 CONSERVEE : ',PMMIN
```

```
NTOTM=MPOS(NV+1)
ZTOTAL=.FALSE.
```

```
C      initialisation du vecteur des poids des modalites et des
C      nombres d'individus concerne
C      -----
```

```
DO 10 I=1,NTOTM
    PM(I)=0.
    NICON(I)=0
10  CONTINUE
```

```
C      calcul des poids des modalites et des nombres d'individus
C      concerne
C      -----
```

```
OPEN(UNIT=NUMCC,FILE=FCOD,DEVICE='DSK')
DO 20 I=1,NI
    READ(NUMCC,*)(NCODI(J),J=1,NUMVR(NV))
    DO 30 K=1,NV
        J=MPOS(K)+NCODI(NUMVR(K))
        PM(J)=PM(J)+PI(I)
        NICON(J)=NICON(J)+1
30    CONTINUE
20    CONTINUE
CLOSE(NUMCC)
```

```
C      liste des modalites de poids trop faible
C      -----
```

```
ZIMP=.TRUE.
DO 40 K=1,NV
    DO 50 J=MPOS(K)+1,MPOS(K+1)
        JJ=J-MPOS(K)
        IF (PM(J).LT.PMMIN) THEN
            IF (ZIMP) THEN
                WRITE(NUMRES,*)
                WRITE(NUMRES,*)'LISTE DES MODALITES DE POIDS TROP
1 FAIBLE : '
                WRITE(NUMRES,*)'-----
1-----'
                ZIMP=.FALSE.
            END IF
            WRITE(NUMRES,*)' VARIABLE : ',K,' -> MODALITE : '
1    ',JJ,' -> POIDS : ',PM(J)
            END IF
50    CONTINUE
40    CONTINUE
```

```
C      nettoyage
C      =====
```

```
C      initialisation
C      -----
```



```

NVG=0
NIG=NI
NMGT=0
NMABAN=0
MPOSVG(1)=0
DO 200 I=1,NV
    MPVG(I)=1
DO 200 J=MPOS(I)+1,MPOS(I+1)
    MPMG(J)=1
200    CONTINUE

C      Pre-elimination des modalites de poids nul et des variables
C      n'ayant plus qu'une modalite.
C      -----

DO 210 K=1,NV
    K1=MPOS(K)+1
    K2=MPOS(K+1)
    NMGV=0
    DO 220 L=K1,K2
        IF (PM(L).LE.1.E-10) THEN
            MPMG(L)=0
            PM(L)=0.
        ELSE
            LL=L
            NMGV=NMGV+1
        END IF
220    CONTINUE
    IF (NMGV.LE.1) THEN
        MPMG(LL)=0
        NMABAN=NMABAN+1
        MPVG(K)=0
    END IF
210    CONTINUE

C      cas ou le nettoyage se fait automatiquement.
C      =====

IF (IQDBT.EQ.IQDBT2) THEN

C      methode par regroupements
C      -----

IF (ICHTR.EQ.ICHTR3) THEN
    WRITE(NUMRES,*)
    ZPGR=.FALSE.
    DO 60 K=1,NV
        IF (MPVG(K).NE.0) THEN
            K1=MPOS(K)+1
            K2=MPOS(K+1)
            DO 80 L=K1,K2
                IF ((MPMG(L).GT.0).AND.(PM(L).LT.PMMIN)) THEN
                    I1=MODGR(L)
                    PAUX=0.
                    WRITE(NUMRES,*)
                    WRITE(NUMRES,*) 'REGROUPEMENT AUTOUR DE LA
1 MODALITE ',I1,' DE LA VARIABLE ',K
                    WRITE(NUMRES,*) ' MODALITES CONCERNEES : '
                    DO 90 L1=K1,K2
                        IF (MODGR(L1).EQ.I1) THEN

```



```

WRITE(NUMRES,*)'                                ',L1-MPOS(K)

PAUX=PAUX+PM(L1)
PM(L1)=0.
MPMG(L1)=-I1
END IF
90 CONTINUE
PM(MPOS(K)+I1)=PAUX
MPMG(MPOS(K)+I1)=1

WRITE(NUMRES,*)' POIDS DE LA MODALITE GROUPE : '
1      ,PM(MPOS(K)+I1)

END IF
80 CONTINUE

C      signalement d'une eventuelle insuffisance de
C      regroupement pour cette variable.

DO 100 L=K1,K2
  IF ((MPMG(L).GT.0).AND.(PM(L).LT.PMMIN)) THEN
    WRITE(NUMTER,*)
    WRITE(NUMTER,*)'MALGRE REGROUPEMENT, POIDS TROP
1 FAIBLE POUR LA'
    WRITE(NUMTER,*)'MODALITE ',L-MPOS(K),
1      ' DE LA VARIABLE ',K
    ZPGR=.TRUE.
  END IF
100 CONTINUE
END IF
60 CONTINUE
ZTOTAL=ZTOTAL.OR.ZPGR

C      cas ou les regroupements sont trop faibles.

IF (ZPGR) THEN
  WRITE(NUMTER,*)
  WRITE(NUMTER,*)'REVOYEZ LES REGROUPEMENT ET RECOMMENCEZ
1 !!'
  RETURN
END IF
END IF

C      methode par elimination des individus.
C      -----

IF (ICHTR.EQ.ICHTR1) THEN
  ZFIN=.FALSE.
  DO 65 WHILE (.NOT.ZFIN)
    ZFIN=.TRUE.
    DO 61 K=1,NV
      IF (MPMG(K).GT.0) THEN
        K1=MPOS(K)+1
        K2=MPOS(K+1)
        DO 81 J=K1,K2
          IF ((MPMG(J).GT.0).AND.(PM(J).LT.PMMIN)) THEN
            ZFIN=.FALSE.
            OPEN(UNIT=NUMCC,FILE=FCOD,DEVICE='DSK')
            NBT=0
            PPLUS=0.
            L=1

```



```

NICAU=NICON(J)
DO 270 WHILE((L.LE.NI).AND.(NBT.LT.NICAU))
  READ(NUMCC,*)(NCODI(I),I=1,NUMVR(NV))
  IF (((J-MPOS(K)).NE.NCODI(NUMVR(K))).OR
1    .(PI(L).LE.1.E-10)) THEN
    L=L+1
  ELSE
    NBT=NBT+1
    NIG=NIG-1
    PPLUS=PPLUS+PI(L)
    DO 280 I=1,NV
      IF (I.NE.K) THEN
        NAU=NCODI(NUMVR(I))+MPOS(I)
        NICON(NAU)=NICON(NAU)-1
        PM(NAU)=PM(NAU)-PI(L)
      END IF
280    CONTINUE
    PI(L)=0.
    L=L+1
  END IF
270  CONTINUE
  CLOSE(NUMCC)
  PAJOUT=PPLUS/NIG
  NICON(J)=0
  PM(J)=0.
  MPMG(J)=0
  DO 290 JJ=1,MPOS(NV+1)
    IF ((NICON(JJ).EQ.0).AND.(MPMG(JJ).GT.0)) THEN
      NMABAN=NMABAN+1
      MPMG(JJ)=0
      PM(JJ)=0.
    ELSE
      PM(JJ)=PM(JJ)+PAJOUT*NICON(JJ)
    END IF
290  CONTINUE
  DO 295 I=1,NI
    IF (PI(I).GT.1.E-10) PI(I)=PI(I)+PAJOUT
295  CONTINUE
  END IF
81  CONTINUE
  END IF
61  CONTINUE
65  CONTINUE
END IF

C      methode par ventilation.
C      -----

IF (ICHTR.EQ.ICHTR2) THEN
  WRITE(NUMRES,*)
  FCOD2='F2.DAT'
  OPEN (UNIT=NUMCC,FILE=FCOD,DEVICE='DSK')
  OPEN (UNIT=NUMCC2,FILE=FCOD2,DEVICE='DSK')
  I=1
  DO 1100 WHILE (I.LE.(NI+NICLA))
    READ(NUMCC,*)(NCODI(K),K=1,NUMVR(NV))
    DO 62 K=1,NV
      IF (MPVG(K).GT.0) THEN
        J=MPOS(K)+NCODI(NUMVR(K))
        IF (PM(J).LT.PMMIN) THEN
          ZPGM=.TRUE.

```



```

DO 84 JL=MPOS(K)+1,MPOS(K+1)
  IF (PM(JL).GE.PMMIN) ZPGM=.FALSE.
  CONTINUE
  IF (ZPGM) GOTO 500
  ZTR=.TRUE.
  NAUX1=1
  NN=MPOS(K+1)-MPOS(K)
  DO 83 WHILE(ZTR)
    NAUX=MOD((K+J+NAUX1),NN)
    IF (NAUX.EQ.0) NAUX=1
    IF (PM(MPOS(K)+NAUX).GE.PMMIN) THEN
      ZTR=.FALSE.
      IF (I.LE.NI) THEN
        PM(MPOS(K)+NAUX)=PM(MPOS(K)+NAUX)+PI(I)
        PM(J)=PM(J)-PI(I)
        MPMG(J)=0
      END IF
      NCODI(NUMVR(K))=NAUX
    ELSE
      NAUX1=NAUX1+1
    END IF
  CONTINUE
83  END IF
  END IF
62  CONTINUE
  WRITE(NUMCC2,'(X,100(I1,X))')(NCODI(K),K=1,NUMVR(NV))
  I=I+1
1100 CONTINUE
  CLOSE(NUMCC)
  CLOSE(NUMCC2)
END IF

C      si nettoyage a la carte.

ELSE

  WRITE(NUMTER,*)
  WRITE(NUMTER,*)'NETTOYAGE A LA CARTE : '
  WRITE(NUMTER,*)'-----'

  ZFIN=.FALSE.
  DO 235 WHILE (.NOT.ZFIN)
    ZFIN=.TRUE.
    DO 231 K=1,NV
      IF (MPVG(K).GT.0) THEN
        K1=MPOS(K)+1
        K2=MPOS(K+1)
        DO 241 J=K1,K2
          IF ((MPMG(J).GT.0).AND.(PM(J).LT.PMMIN)) THEN

            WRITE(NUMTER,*)
            WRITE(NUMTER,*)'VARIABLE : ',K
1            , ' POIDS TROP FAIBLE DE LA MODALITE : '
2            ,J-MPOS(K)
            WRITE(NUMTER,*)' -> POIDS ',PM(J)
1            , ' -> NB D'INDIVIDUS CONCERNES ',NICON(J)
            WRITE(NUMTER,*)
            CALL SAVA1(NUMTER,NUMTER,NUMTER,IREV,0,1,50
1            ,116,117,ZP)
            IF (IREV.EQ.1) THEN
              WRITE(NUMTER,*)

```



```

WRITE(NUMTER,*)'VARIABLE :',K
DO 251 JJ=K1,K2
  IF (MPMG(JJ).GT.0) THEN
    WRITE(NUMTER,*)'--> MODALITE ',JJ-MPOS(K)
    WRITE(NUMTER,*)'          POIDS ',PM(JJ)
    ',', NB D''INDIVIDUS CONCERNES ',NICON(JJ)
  END IF
251 CONTINUE
END IF
WRITE(NUMTER,*)
WRITE(NUMTER,*)'MODALITE :',J-MPOS(K)
CALL SAVA1(NUMTER,NUMTER,NUMTER,ICHOIX,0,1,51
1      ,118,119,ZP)

C      traitement par jumelage.

IF (ICHOIX.EQ.1) THEN
  ZL=.TRUE.
  DO 261 WHILE (ZL)
    WRITE(NUMTER,*)'NUMERO DE LA MODALITE ?'
    CALL LEC1(NUMTER,NJUM,ZL)
    IF (ZL) THEN
      WRITE(NUMTER,*)
      WRITE(NUMTER,*)'ERREUR LORS DE LA LECTURE
1 DE CE NUMERO !!'
      ZL=.TRUE.
    ELSE
      IF ((NJUM.LT.1).OR.(NJUM.GT.(K2-K1+1)).OR.
1      (NJUM.EQ.(J-K1+1))) THEN
        WRITE(NUMTER,*)'NUMERO INVALIDE !!'
        ZL=.TRUE.
      ELSE
        IF (MPMG(MPOS(K)+NJUM).LE.0) THEN
          WRITE(NUMTER,*)'NUMERO INVALIDE !!'
          ZL=.TRUE.
        ELSE
          ZL=.FALSE.
        END IF
      END IF
    END IF
  END IF
261 CONTINUE

C      modifications dues au regroupement.

NICON(MPOS(K)+NJUM)=NICON(MPOS(K)+NJUM)+
1 NICON(J)
PM(MPOS(K)+NJUM)=PM(MPOS(K)+NJUM)+PM(J)
IF (PM(MPOS(K)+NJUM).LT.PMMIN) ZFIN=.FALSE.
PM(J)=0.
NICON(J)=0
MPMG(J)=-NJUM
DO 264 JL=K1,K2
  IF (MPMG(JL).EQ.-(J-MPOS(K))) MPMG(JL)=-NJUM
264 CONTINUE
WRITE(NUMRES,*)
WRITE(NUMRES,*)'VARIABLE ',K,': FUSION DE LA
1 MODALITE ',J-MPOS(K), ' AVEC LA MODALITE ',NJUM

C      traitement par elimination des individus
C      concernees.

```



```

ELSE
  OPEN(UNIT=NUMCC,FILE=FCOD,DEVICE='DSK')
  NBT=0
  PPLUS=0.
  L=1
  NICAU=NICON(J)
  DO 271 WHILE ((L.LE.NI).AND.(NBT.LT.NICAU))
    READ(NUMCC,*)(NCODI(I),I=1,NUMVR(NV))
    IF (((J-MPOS(K)).NE.NCODI(NUMVR(K))).AND.
      1 (-MPMG(MPOS(K)+NCODI(NUMVR(K))).NE.
      2 (J-MPOS(K))).OR.(PI(L).LE.1.E-10)) THEN
      L=L+1
    ELSE
      NBT=NBT+1
      NIG=NIG-1
      PPLUS=PPLUS+PI(L)
      DO 281 I=1,NV
        IF (I.NE.K) THEN
          1 NAU=NCODI(NUMVR(I))+MPOS(I)
          IF (MPMG(NAU).LT.0) NAU=MPOS(I)-MPMG
            (NAU)
          PM(NAU)=PM(NAU)-PI(L)
          NICON(NAU)=NICON(NAU)-1
        END IF
      281 CONTINUE
      PI(L)=0.
      L=L+1
    END IF
  271 CONTINUE
  CLOSE(NUMCC)
  PAJOUT=PPLUS/NIG
  NICON(J)=0
  PM(J)=0.
  MPMG(J)=0
  DO 289 JJ=K1,K2
    IF (MPMG(JJ).EQ.-(J-MPOS(K))) MPMG(JJ)=0
  289 CONTINUE
  DO 291 JJ=1,MPOS(NV+1)
    IF ((NICON(JJ).EQ.0).AND.(MPMG(JJ).GT.0)) THEN
      NMABAN=NMABAN+1
      MPMG(JJ)=0
      PM(JJ)=0.
    ELSE
      PM(JJ)=PM(JJ)+PAJOUT*NICON(JJ)
    END IF
  291 CONTINUE
  DO 288 I=1,NI
    IF (PI(I).GT.1.E-10) PI(I)=PI(I)+PAJOUT
  288 CONTINUE
  END IF
END IF
241 CONTINUE
END IF
231 CONTINUE
235 CONTINUE
END IF

C traitement general

DO 300 K=1,NV
  IF (MPVG(K).GT.0) THEN

```



```

K1=MPOS(K)+1
K2=MPOS(K+1)
NMGV=0
DO 310 L=K1,K2
  IF (MPMG(L).LE.0) THEN
    NMABAN=NMABAN+1
  ELSE
    NMGV=NMGV+1
    NMGT=NMGT+1
    MPMG(L)=NMGT
    DO 305 I=K1,K2
      IF (MPMG(I).EQ.-(L-MPOS(K))) MPMG(I)=- (100+NMGT)
305    CONTINUE
    LL=L
  END IF
310 CONTINUE

C      traitement de la variable en fonction du poids de la
C      modalite non-repondu.

  IF (PM(MPOS(K+1)).GT.PNRV) THEN
    NMGT=NMGT-NMGV
    DO 320 I=K1,K2
      MPMG(I)=0
320    CONTINUE
    NMABAN=NMABAN+NMGV
    MPVG(K)=-1
  ELSE

C      traitement de la variable en fonction du nombre de
C      modalites restant.

    IF (NMGV.GT.1) THEN
      NVG=NVG+1
      MPVG(K)=NVG
      MPOSVG(NVG+1)=MPOSVG(NVG)+NMGV
    ELSE
      NMGT=NMGT-1
      MPMG(LL)=0
      NMABAN=NMABAN+1
      MPVG(K)=0
    END IF
  END IF
300 CONTINUE

C      liste des numeros de variables eliminees pour pnrp trop
C      grand.

ZIMP=.TRUE.
DO 330 K=1,NV
  IF (MPVG(K).EQ.-1) THEN
    IF (ZIMP) THEN
      WRITE(NUMRES,*)
      WRITE(NUMRES,*) 'VARIABLES ELIMINEES POUR CAUSE DE POIDS
1 TROP'
      WRITE(NUMRES,*) 'IMPORTANT DE LA MODALITE NON-REPONDU'
      WRITE(NUMRES,*) '-----'
1-----
      ZIMP=.FALSE.
    END IF
  
```



```

        WRITE(NUMRES,*)'      VARIABLE ',K
    END IF
330    CONTINUE

C      liste des numeros de variables eliminees pour nombre
C      insuffisant de modalites.

    ZIMP=.TRUE.
    DO 340 K=1,NV
        IF (MPVG(K).EQ.0) THEN
            IF (ZIMP) THEN
                WRITE(NUMRES,*)
                WRITE(NUMRES,*)'VARIABLES ELIMINEES POUR CAUSE DE
1 NOMBRE '
                WRITE(NUMRES,*)'DE MODALITES EGAL A 1'
                WRITE(NUMRES,*)'-----
1-----'
                ZIMP=.FALSE.
            END IF
            WRITE(NUMRES,*)'      VARIABLE ',K
        END IF
340    CONTINUE

C      changement ou non dans le fichier des codes condensés.

    IF (NMABAN.EQ.0) THEN
        FCOD1=FCOD
    ELSE
        IF (ICHTR.EQ.ICHTR2) THEN
            OPEN (UNIT=NUMCC2,FILE=FCOD2,DEVICE='DSK')
        ELSE
            OPEN (UNIT=NUMCC,FILE=FCOD,DEVICE='DSK')
        END IF
        FCOD1='F1.DAT'

C      mise a jour du fichier des codes condensés.

        IF ((NVG.GE.1).AND.(NIG.GE.1)) THEN
            OPEN(UNIT=NUMCC1,FILE=FCOD1,DEVICE='DSK')
            DO 140 I=1,NI
                IF (ICHTR.EQ.ICHTR2) THEN
                    READ(NUMCC2,*)(NCODI(K),K=1,NUMVR(NV))
                ELSE
                    READ(NUMCC,*)(NCODI(K),K=1,NUMVR(NV))
                END IF
                IF (PI(I).GT.0.) THEN
                    DO 145 K=1,NV
                        K1=MPVG(K)
                        IF (K1.GT.0) THEN
                            LA=MPOS(K)+NCODI(NUMVR(K))
                            LN=MPMG(LA)
                            IF (LN.LE.0) THEN
                                NK=-(LN+100)-MPOSVG(K1)
                            ELSE
                                NK=LN-MPOSVG(K1)
                            END IF
                            NCODI(K1)=NK
                        END IF
                    END IF
145                CONTINUE
                WRITE(NUMCC1,'(X,100(I1,X))')(NCODI(J),J=1,NVG)
            END IF

```


CONTINUE

IF (NICLA.GT.0) THEN

DO 1040 I=1,NICLA

IF (ICHTR.EQ.ICHTR2) THEN

READ(NUMCC2,*)(NCODI(K),K=1,NUMVR(NV))

ELSE

READ(NUMCC,*)(NCODI(K),K=1,NUMVR(NV))

END IF

DO 1045 K=1,NV

K1=MPVG(K)

IF (K1.GT.0) THEN

LA=MPOS(K)+NCODI(NUMVR(K))

LN=MPMG(LA)

IF (LN.LT.0) THEN

NK=-(LN+100)-MPOSVG(K1)

ELSE

IF (LN.EQ.0) THEN

NN=MPOSVG(K1+1)-MPOSVG(K1)

NK=MOD((K+I+LA),NN)

IF (NK.EQ.0) NK=1

ELSE

NK=LN-MPOSVG(K1)

END IF

END IF

NCODI(K1)=NK

END IF

1045

CONTINUE

WRITE(NUMCC1,'(X,100(I1,X))')(NCODI(J),J=1,NVG)

1040

CONTINUE

END IF

IF (ICHTR.EQ.ICHTR2) THEN

CLOSE(NUMCC2)

ELSE

CLOSE(NUMCC)

END IF

CLOSE(NUMCC1)

END IF

END IF

C

mise a Jour de NGR et de NVIGR.

IF (NIG.LT.NI) THEN

I1=1

I2=1

IMIN=0

NBIGR=0

DO 401 I=1,NI

IF (PI(I).GT.0.) THEN

NBIGR=NBIGR+1

END IF

IF (NVIGR(I1).LE.(I-IMIN)) THEN

IF (NBIGR.EQ.0) THEN

NGR=NGR-1

IMIN=I

I1=I1+1

NBIGR=0

ELSE

NVIGR(I2)=NBIGR

IMIN=I

I1=I1+1

I2=I2+1


```

        NBIGR=0
        END IF
        END IF
401    CONTINUE
    END IF

C      compactage du vecteur des poids des individus et liste des
C      individus supprimees.

    IF (NIG.LT.NI) THEN
        WRITE(NUMRES,*)
        WRITE(NUMRES,*)'LISTE DES INDIVIDUS SUPPRIMES :/'
        WRITE(NUMRES,*)'-----/'
        WRITE(NUMRES,*)
        I1=0
        DO 400 I=1,NI
            IF (PI(I).GT.0.) THEN
                I1=I1+1
                PI(I1)=PI(I)
            ELSE
                WRITE(NUMRES,*)'  INDIVIDU ',I
            END IF
400        CONTINUE
        DO 410 I=NIG+1,NI
            PI(I)=0.
410        CONTINUE
    END IF

C      mise a jour ou non des libelles.

    IF (ILIB.EQ.ILIB2) THEN
        IF ((NMABAN.GT.0).AND.(NVG.GE.1)) THEN

C          mise a jour des vecteurs des libelles

            IF (NVG.LT.NV) THEN
                DO 430 I=1,NV
                    IF (MPVG(I).GT.0) LIBVAR(MPVG(I))=LIBVAR(I)
430                CONTINUE
            END IF
            DO 440 I=1,MPOS(NV+1)
                IF (MPMG(I).GT.0) LIBMOD(MPMG(I))=LIBMOD(I)
440            CONTINUE
        END IF
    END IF

C      compactage du vecteur des poids des modalites.

    IF (NMABAN.GT.0) THEN
        DO 480 I=1,MPOS(NV+1)
            IF (MPMG(I).GT.0) PM(MPMG(I))=PM(I)
480        CONTINUE
    END IF

C      liste des poids des modalites apres nettoyage.

    IF ((IPMN.EQ.IPMN2).AND.(NVG.GE.1)) THEN
        WRITE(NUMRES,*)
        WRITE(NUMRES,*)'POIDS DES MODALITES APRES NETTOYAGE/'
        WRITE(NUMRES,*)'-----/'
        DO 460 K=1,NVG

```



```

        WRITE(NUMRES,*)
        WRITE(NUMRES,*)' VARIABLE : ',K
        DO 470 J=MPOSVG(K)+1,MPOSVG(K+1)
            JJ=J-MPOSVG(K)
            WRITE(NUMRES,*)'    MODALITE ',JJ,' --> POIDS ',PM(J)
470      CONTINUE
460      CONTINUE
        END IF

C      mise a jour de mpos, nv, ni

        IF ((NMABAN.GT.0).AND.(NVG.GE.1)) THEN
            DO 420 I=1,NVG+1
                MPOS(I)=MPOSVG(I)
420          CONTINUE
            END IF
            NV=NVG
            NI=NIG
            NTOTM=MPOS(NV+1)
            WRITE(NUMRES,*)

            RETURN

500      WRITE(NUMTER,*)'VENTILATION IMPOSSIBLE, CHOISISSEZ UNE'
            WRITE(NUMTER,*)'AUTRE METHODE ET RECOMMENCEZ !!'
            ZTOTAL=.TRUE.
            RETURN
        END

C+++++
@

```



```

20      CONTINUE

C      calcul du tableau de burt croisant les modalites des varia-
C      bles avec celles de u.

      OPEN(UNIT=NUMCC1,FILE=FCOD1,DEVICE='DSK')
      IMIN=0
      NMGR=1
      DO 30 I=1,NI
        IF (NVIGR(NMGR).LT.(I-IMIN)) THEN
          IMIN=I-1
          NMGR=NMGR+1
        END IF
        READ(NUMCC1,*)(NCODI(K),K=1,NV)
        DO 40 K1=1,NV
          J=MPOS(K1)+NCODI(K1)
          BURTY(J,NMGR)=BURTY(J,NMGR)+PI(I)
40      CONTINUE
30      CONTINUE
      CLOSE(NUMCC1)

C      calcul du tableau de burt croisant les modalites des
C      variables

      OPEN (UNIT=NBURTA,FILE=FBURTA,DEVICE='DSK')
      DO 21 K1=1,NV
        NMK1=MPOS(K1+1)-MPOS(K1)

C      initialisation du bloc correspondant a la variable K1

        DO 22 I=1,NMK1
          DO 22 J=1,MPOS(NV+1)
            BURT(I,J)=0.
22      CONTINUE

C      calcul des elements de ce bloc

        OPEN (UNIT=NUMCC1,FILE=FCOD1,DEVICE=DSK)
        DO 23 I=1,NI
          READ(NUMCC1,*)(NCODI(K),K=1,NV)
          J=NCODI(K1)
          DO 24 K2=1,NV
            L=MPOS(K2)+NCODI(K2)
            BURT(J,L)=BURT(J,L)+PI(I)
24      CONTINUE
23      CONTINUE
        CLOSE (NUMCC1)

C      sauvegage de ce bloc dans le fichier FBURTA

        DO 25 J1=1,NMK1
          WRITE(NBURTA,'(900(D25.18,X))')(BURT(J1,J2),J2=1
1      ,MPOS(NV+1))
25      CONTINUE

21      CONTINUE
      CLOSE (NBURTA)

C      calcul des coefficients de tschurrow du premier ordre entre
C      les variables.

```



```

CALL RINTAB(TSCHX,NV,1,1,1.DO)
IF (NV.GT.1) THEN
  OPEN (UNIT=NBURTA,FILE=FBURTA,DEVICE='DSK')
  DO 53 I1=1,MPOS(2)
    READ(NBURTA,*)
53  CONTINUE
    DO 50 I=2,NV
      MI=MPOS(I+1)-MPOS(I)
      DO 52 II1=1,MI
        READ(NBURTA,*)(BURT(II1,II2),II2=1,MPOS(I))
52  CONTINUE
        CALL RINTAB(TSCHX,NV,I,I,1.DO)
        DO 50 J=1,I-1
          MJ=MPOS(J+1)-MPOS(J)
          DO 60 I1=1,MI
            TETAK(I1)=0.
            DO 80 J1=1,MJ
              TETAK(I1)=TETAK(I1)+BURT(I1,MPOS(J)+J1)
80  CONTINUE
60  CONTINUE
          DO 70 J1=1,MJ
            TETAL(J1)=0.
            DO 90 I1=1,MI
              TETAL(J1)=TETAL(J1)+BURT(I1,MPOS(J)+J1)
90  CONTINUE
70  CONTINUE
          SOM=-1.
          DO 100 K=1,MI
            DO 100 L=1,MJ
              SOM=SOM+BURT(K,MPOS(J)+L)*BURT(K,MPOS(J)+L)/
1      (TETAK(K)*TETAL(L))
100 CONTINUE
          CALL RINTAB(TSCHX,NV,I,J,SOM/SQRT(FLOAT((MI-1)*(MJ-1))))
50 CONTINUE
  CLOSE (NBURTA)
END IF

C      impression du tableau des coefficients de tschuprow du
C      premier ordre.

IF (ICT1.EQ.ICT12) THEN
  WRITE(NUMRES,*)
  WRITE(NUMRES,*) 'TABLEAU DES COEFFICIENTS DE TSCHUPROW
1 DU PREMIER ORDRE'
  WRITE(NUMRES,*) 'ENTRE LES VARIABLES (TRIANGLE INFERIEUR)'
  WRITE(NUMRES,*) '-----'
1  '-----'
  WRITE(NUMRES,*)
  DO 51 I=1,NV
    WRITE(NUMRES, '(X,100(X,D14.7))')(ROUTAB(TSCHX,NV,I,J)
1    ,J=1,I)
51 CONTINUE
END IF

C      calcul des coefficients de tschuprow du premier ordre avec u.

MY=NGR
DO 110 I=1,NV
  MI=MPOS(I+1)-MPOS(I)
  DO 120 I1=1,MI
    TETAK(I1)=0.

```



```

        DO 130 J1=1,MY
            TETAK(I1)=TETAK(I1)+BURTY(MPOS(I)+I1,J1)
130      CONTINUE
120      CONTINUE
        DO 140 J1=1,MY
            TETAL(J1)=0.
            DO 150 I1=1,MI
                TETAL(J1)=TETAL(J1)+BURTY(MPOS(I)+I1,J1)
150      CONTINUE
140      CONTINUE
        SOM=-1.
        DO 160 K=1,MI
            DO 160 L=1,MY
                SOM=SOM+BURTY(MPOS(I)+K,L)*BURTY(MPOS(I)+K,L)
1      /((TETAK(K)*TETAL(L))
160      CONTINUE
        TSCHY(I)=SOM/SQRT(FLOAT((MY-1)*(MI-1)))
110      CONTINUE

C      impression des coefficients de tschuprow du premier ordre
C      avec y.

        IF (ICTY1.EQ. ICTY12) THEN
            WRITE(NUMRES,*)
            WRITE(NUMRES,*) 'VECTEUR DES COEFFICIENTS DE TSCHUPROW DU
1      PREMIER ORDRE AVEC Y'
            WRITE(NUMRES,*) '-----'
            WRITE(NUMRES,*)
            WRITE(NUMRES, '(X,100(X,D14.7))' ) (TSCHY(I),I=1,NV)
        END IF

C      initialisation du vecteur d'etat des variables.

        DO 115 I=1,NV
            MPVS(I)=0
115      CONTINUE

C      elimination des variables redondantes.

        DO 170 I=2,NV
            IF (MPVS(I).EQ.0) THEN
                DO 180 J=1,I-1
                    IF (MPVS(J).EQ.0) THEN
                        IF (ROUTAB(TSCHX,NV,I,J).GE.SR) MPVS(I)=- (J+NV)
                    END IF
180      CONTINUE
                END IF
170      CONTINUE

C      liste des variables eliminees pour redondance.

        ZIMP=.TRUE.
        DO 190 I=1,NV
            IF (MPVS(I).LT.0) THEN
                IF (ZIMP) THEN
                    WRITE(NUMRES,*)
                    WRITE(NUMRES,*) 'VARIABLES ELIMINEES POUR REDONDANCE'
                    WRITE(NUMRES,*) '-----'
                    WRITE(NUMRES,*)
                    ZIMP=.FALSE.

```



```

        END IF
        WRITE(NUMRES,*)'VARIABLE ',I
1      , ' REDONDANTE AVEC LA VARIABLE ',--MPVS(I)--NV
        END IF
190    CONTINUE

C      initialisation de la selection.

        WRITE(NUMRES,*)
        WRITE(NUMRES,*)'SELECTION DES VARIABLES DISCRIMINANTES : '
        WRITE(NUMRES,*)'===== '

        CTM=0.
        NVSEFF=0
        ZSELEC=.TRUE.
        NPER1=1
        NPER2=1
        NPAS=0

C      selection.

        DO 400 WHILE(ZSELEC)
            NPAS=NPAS+1
            WRITE(NUMRES,*)
            WRITE(NUMRES,*)'PAS : ',NPAS,' : '
            WRITE(NUMRES,*)'-----'

C      choix de la variable.

            TSCMAX=0.
            IVS=0.
            DO 390 K=1,NV
                IF (MPVS(K).EQ.0) THEN
                    IF (ABS(TSCHY(K)).GE.TSCMAX) THEN
                        TSCMAX=ABS(TSCHY(K))
                        IVS=K
                    END IF
                END IF
            CONTINUE
390        NVSEFF=NVSEFF+1
            MPVS(IVS)=NVSEFF

C      impression de tscmax.

            IF (ICTMX.EQ.ICTMX2) THEN
                WRITE(NUMRES,*)
                WRITE(NUMRES,*)'COEFFICIENT DE TSCHUPROW MAXIMUM
1      AVEC Y : ',TSCMAX
            END IF

C      impression de ivs.

            WRITE(NUMRES,*)
            WRITE(NUMRES,*)'INDICE DE LA VARIABLE SELECTIONNEE : '
1      ,IVS

C      calcul du coefficient de tschuprow multiple.

            CTM1=SQRT(1.-CTM*(1.-CTM)*(1.-TSCMAX*TSCMAX))

C      impression du coefficient de tschuprow multiple et de sa

```


C croissance.

```
IF (ICTMP.EQ.ICTMP2) THEN
  WRITE(NUMRES,*)
  WRITE(NUMRES,*)'COEFFICIENT DE TSCHUPROW MULTIPLE : '
1  ,CTM1
  WRITE(NUMRES,*)'CROISSANCE % A L'ETAPE PRECEDENTE : '
1  ,CTM1-CTM
END IF
```

C methode d'arret par atm.

```
IF (IMAR.EQ.IMAR2) THEN
  IF (CTM1.LT.(CTM+QTM)) THEN
    ZSELEC=.FALSE.
    IF (NPAS.GT.1) THEN
      MPVS(IVS)=0
      NVSEFF=NVSEFF-1
    END IF
    WRITE(NUMRES,*)
    WRITE(NUMRES,*)'ARRET DE LA SELECTION CAR LE CTM NE
1 CROIT PLUS SUFFISAMMENT'
    WRITE(NUMRES,*)
    IF (NPAS.GT.1) WRITE(NUMRES,*)'ABANDON DE LA VARIABLE'
1  ,IVS
    ELSE
      CTM=CTM1
    END IF
  END IF
```

C methode d'arret par nombre fixe de variables a
C selectionner.

```
ELSE
  IF (NVSEFF.GE.NVS) THEN
    ZSELEC=.FALSE.
    WRITE(NUMRES,*)
    WRITE(NUMRES,*)'ARRET DE LA SELECTION CAR LE NOMBRE DE
1 VARIABLES'
    WRITE(NUMRES,*)'A SELECTIONNER EST ATTEINT'
    ELSE
      CTM=CTM1
    END IF
  END IF
END IF
```

C eventuel arret premature si tscmax=1

```
IF (ZSELEC) THEN
  IF (TSCMAX.GE.SR) THEN
    WRITE(NUMRES,*)
    WRITE(NUMRES,*)'ARRET DE LA SELECTION CAR TSCMAX >= SR'
    ZSELEC=.FALSE.
  END IF
END IF
```

C elimination et impression des variables redondantes avec la
C variable selectionnee.

```
IF (ZSELEC) THEN
  ZIMP=.TRUE.
  IF (NV.GT.1) THEN
    DO 380 K=1,NV
```



```

      IF (MPVS(K).EQ.0) THEN
        IF (ABS(ROUTAB(TSCHX,NV,K,IVS)).GE.SR) THEN
          MPVS(K)=-IVS
          IF (ZIMP) THEN
            WRITE(NUMRES,*)
            WRITE(NUMRES,*)'VARIABLES ELIMINEES POUR CAUSE
1 DE COEFFICIENT DE'
            WRITE(NUMRES,*)'TSCHUPROW D'ORDRE K ~ 1 AVEC
1 LA VARIABLE SELECTIONNEE'
            WRITE(NUMRES,*)'-----
1-----'
            ZIMP=.FALSE.
          END IF
          WRITE(NUMRES,*)' VARIABLE ',K
        END IF
      END IF
380    CONTINUE
    END IF
  END IF

C      eventuel arrat s'il n'y a plus de variables selectionna-
C      bles.

  IF (ZSELEC) THEN
    I=0
    K=1
    DO 370 WHILE ((I.EQ.0).AND.(K.LE.NV))
      IF (MPVS(K).EQ.0) I=I+1
      K=K+1
370    CONTINUE
    IF (I.EQ.0) THEN
      WRITE(NUMRES,*)
      WRITE(NUMRES,*)'ARRET DE LA SELECTION CAR IL N'Y A
1 PLUS DE'
      WRITE(NUMRES,*)'VARIABLE SELECTIONNABLE'
      ZSELEC=.FALSE.
    END IF
  END IF

  IF (ZSELEC) THEN

C      transformation du tableau des coefficients de tschuprow
C      entre les variables.

    DO 360 K1=2,NV
      IF (MPVS(K1).EQ.0) THEN
        DO 350 K2=1,K1-1
          IF (MPVS(K2).EQ.0) THEN
            T1=ROUTAB(TSCHX,NV,K1,K2)
            T2=ROUTAB(TSCHX,NV,K1,IVS)
            T3=ROUTAB(TSCHX,NV,K2,IVS)
            TAUX=(T1-T2*T3)/SQRT((1,-T2*T2)*(1,-T3*T3))
            CALL RINTAB(TSCHX,NV,K1,K2,TAUX)
          END IF
350        CONTINUE
      END IF
360    CONTINUE

C      impression de ce tableau.

    IF (ICTF.GT.0) THEN

```



```

      IF (NVSEFF.EQ.(ICTF*NPER1)) THEN
        NPER1=NPER1+1
        WRITE(NUMRES,*)
        WRITE(NUMRES,*)'TABLEAU DES COEFFICIENTS DE TSCHUPROW
1 ENTRE LES VARIABLES'
        WRITE(NUMRES,*)'APRES TRANSFORMATION (TRIANGLE
1 INFERIEUR)'
        WRITE(NUMRES,*)'-----
1-----'

        DO 340 I=1,NV
          WRITE(NUMRES,('(X,100(X,D14.7))')(ROUTAB(TSCHX
1          ,NV,I,J),J=1,I)
340      CONTINUE
        END IF
      END IF

C      transformation du vecteur des coefficients de tschuprow
C      avec v.

      DO 330 K1=1,NV
        IF (MPVS(K1).EQ.0) THEN
          T1=TSCHY(K1)
          T2=TSCHY(IVS)
          T3=ROUTAB(TSCHX,NV,K1,IVS)
          TAUX=(T1-T2*T3)/SQRT((1.-T2*T2)*(1.-T3*T3))
          TSCHY(K1)=TAUX
        END IF
330      CONTINUE

C      impression de ce tableau.

      IF (ICTYF.GT.0) THEN
        IF (NVSEFF.EQ.(ICTYF*NPER2)) THEN
          NPER2=NPER2+1
          WRITE(NUMRES,*)
          WRITE(NUMRES,*)'VECTEUR DES COEFFICIENTS DE TSCHUPROW
1 AVEC Y'
          WRITE(NUMRES,*)'-----
1-----'
          WRITE(NUMRES,('(X,100(D14.7,X))')(TSCHY(J),J=1,NV)
          END IF
        END IF
      END IF
400      CONTINUE

C      cloture de la selection.
C      -----

C      mise a jour du fichier des codes condensés.

      OPEN(UNIT=NUMCC1,FILE=FCOD1,DEVICE='DSK')
      OPEN(UNIT=NUMCCN,FILE=FCODN,DEVICE='DSK')
      DO 300 I=1,NI+NICLA
        READ(NUMCC1,*)(NCODI(K),K=1,NV)
        DO 310 J=1,NV
          IF (MPVS(J).GT.0) NCODIN(MPVS(J))=NCODI(J)
310      CONTINUE
        WRITE(NUMCCN,('(100(I1,X))')(NCODIN(K),K=1,NVSEFF)
300      CONTINUE
      CLOSE(NUMCC1)

```


CLOSE(NUMCCN)

C nouveau vecteur de position des modalites.

MPOS1(1)=0

DO 249 L=1,NVSEFF

 Z1=.FALSE.

 IV=0

 DO 250 WHILE(.NOT.Z1)

 IV=IV+1

 IF (MPVS(IV).EQ.L) THEN

 Z1=.TRUE.

 MPOS1(L+1)=MPOS1(L)+MPVS(IV+1)-MPVS(IV)

 END IF

250 CONTINUE

249 CONTINUE

C mise a jour du fichier des libelles.

IF (ILIB.EQ.ILIB2) THEN

 OPEN(UNIT=NUMFLN,FILE=FLIBN,DEVICE='DSK')

 DO 290 K=1,NV

 IF (MPVS(K).GT.0) THEN

 LIBV1(MPVS(K))=LIBVAR(K)

 DO 280 J=1,MPVS(K+1)-MPVS(K)

 J1=MPOS1(K)+J

 LIBM1(J1)=LIBMOD(MPOS(K)+J)

280 CONTINUE

 END IF

290 CONTINUE

 DO 270 K=1,NVSEFF

 K1=MPOS1(K)+1

 K2=MPOS1(K+1)

 NL=K2-K1+1

 WRITE(NUMFLN,'(A10,X,I,/,9(A4,X))') LIBV1(K),NL

1 ,(LIBM1(J),J=K1,K2)

270 CONTINUE

 CLOSE(NUMFLN)

END IF

C mise a jour du fichier contenant le tableau de burt

C croisant les modalites des variables explicatives

C entre elles

OPEN (UNIT=NBURT,FILE=FBURT,DEVICE='DSK')

OPEN (UNIT=NBURTA,FILE=FBURTA,DEVICE='DSK')

DO 600 K=1,NVSEFF

 L=0

 ZTV=.FALSE.

 DO 610 WHILE(.NOT.ZTV)

 L=L+1

 IF (MPVS(L).EQ.K) ZTV=.TRUE.

610 CONTINUE

 IF (L.GT.1) THEN

 DO 601 LL1=1,MPVS(L)

 READ(NBURTA,*)

601 CONTINUE

 END IF

 M1=MPVS(L+1)-MPVS(L)

 DO 420 J=1,M1

 READ(NBURTA,*)(BURT(1,JJ1),JJ1=1,MPVS(NV+1))


```

DO 500 KK=1,NVSEFF
  L=0
  ZTV=,FALSE.
  DO 510 WHILE (,NOT.ZTV)
    L=L+1
    IF (MPVS(L),EQ,KK) ZTV=,TRUE.
510    CONTINUE
    DO 520 JJ=MPOS1(KK)+1,MPOS1(KK+1)
      B1(JJ)=BURT(1,JJ-MPOS1(KK)+MPOS(L))
520    CONTINUE
500    CONTINUE
    WRITE(NBURT,'(900(D25,18,X))')(B1(I)
1    ,I=1,MPOS1(NVSEFF+1))
420    CONTINUE
    REWIND NBURTA
600    CONTINUE

CLOSE(NBURT)
CLOSE(NBURTA)

C    mise a jour du tableau de burt croisant les modelites
C    des variables explicatives avec celles de y

OPEN (UNIT=NBURTY,FILE=FBURTY,DEVICE='DSK')
DO 705 K1=1,NGR
  DO 700 K=1,NVSEFF
    L=0
    ZTV=,FALSE.
    DO 710 WHILE (,NOT.ZTV)
      L=L+1
      IF (MPVS(L),EQ,K) ZTV=,TRUE.
710    CONTINUE
      DO 720 J=MPOS1(K)+1,MPOS1(K+1)
        B1(J)=BURTY(J-MPOS1(K)+MPOS(L),K1)
720    CONTINUE
700    CONTINUE
      WRITE(NBURTY,'(900(D25,18,X))')(B1(J),J=1,MPOS1(NVSEFF+1))
705    CONTINUE
  CLOSE(NBURTY)

C    formation du fichier des donnees pour le second programme

OPEN (UNIT=NFDON2,FILE=FDON2,DEVICE='DSK')
WRITE(NFDON2,'(I2)')NVSEFF
WRITE(NFDON2,'(4I(I3,X))')(MPOS1(I),I=1,NVSEFF+1)
WRITE(NFDON2,'(I3)')NI
WRITE(NFDON2,'(I)')NICLA
WRITE(NFDON2,'(4000(D25,18,X))')(PI(I),I=1,NI)
WRITE(NFDON2,'(I1)')NGR
WRITE(NFDON2,'(9(I3,X))')(NVIGR(I),I=1,NGR)
WRITE(NFDON2,'(A10)')FBURT
WRITE(NFDON2,'(A10)')FBURTY
WRITE(NFDON2,'(A10)')FCODN
WRITE(NFDON2,'(I1)')NOUT
IF (NOUT,EQ,NOUT3) WRITE(NFDON2,'(A10)')FRES
WRITE(NFDON2,'(D14,7)')VLPMIN
WRITE(NFDON2,'(D14,7)')PDISCU
CLOSE(NFDON2)

C    impressions complementaires

```

```

WRITE(NUMRES,*)'NOM DU FICHIER DES CODAGES CONDENSES APRES
1 SELECTION :',FCODN
IF (ILIB.EQ.ILIB2) THEN
  WRITE(NUMRES,*)'NOM DU FICHIER DES LIBELLES APRES
1 SELECTION :',FLIBN
END IF
WRITE(NUMRES,*)'NOM DU FICHIER CONTENANT LE TABLEAU DE
1 BURT ENTRE'
WRITE(NUMRES,*)'LES VARIABLES :',FBURT
WRITE(NUMRES,*)'NOM DU FICHIER CONTENANT LE TABLEAU DE
1 BURT ENTRE'
WRITE(NUMRES,*)'LES VARIABLES ET Y :',FBURTY
WRITE(NUMRES,*)'NOM DU FICHIER DE DONNEES POUR LE
1 DEUXIEME PROGRAMME :',FDON2

RETURN
END

```

C+++++

```

SUBROUTINE RINTAB(TAB,NDIM,I,J,VAL)

IMPLICIT DOUBLE PRECISION (A-H,O-Y)

DIMENSION TAB((NDIM*(NDIM+1))/2)

IA=I
JA=J
IF (IA.LT.JA) THEN
  IAU=IA
  IA=JA
  JA=IAU
END IF
IAUX=IA*(IA-1)*.5+JA
TAB(IAUX)=VAL
RETURN
END

```

C+++++

```

DOUBLE PRECISION FUNCTION ROUTAB(TAB,NDIM,I,J)

IMPLICIT DOUBLE PRECISION (A-H,O-Y)

DIMENSION TAB((NDIM*(NDIM+1))/2)

IA=I
JA=J
IF (IA.LT.JA) THEN
  IAU=IA
  IA=JA
  JA=IAU
END IF
IAUX=IA*(IA-1)*.5+JA
ROUTAB=TAB(IAUX)
RETURN
END

```

C+++++
@


```

*****C
C*
C*          BLOC DES ROUTINES DE SAISIE ET VALIDATION
C*
C*****C

```

C Pour chaque routine de ce bloc :

```

C  NUMTER  : numero d'unité des terminaux
C  NR      : numero d'unité pour la lecture
C  NW      : numero d'unité pour l'affichage des messages
C  MSGT     : numero du message de demande a afficher si on
C             introduit les donnees au terminal
C  MSGE... : numeros de messages d'erreur a afficher lorsqu'il
C             y a un probleme
C  ZE      : Booléen valant false si tout s'est passe correc-
C             tement et true sinon ( s'il y a eu un probleme
C             lors de la lecture ou lors de la validation et que
C             la saisie se fait a partir d'un fichier ). ZE sera
C             toujours a false si la saisie se fait a partir
C             d'un terminal.
C  ZC      : Booléen valant true tant qu'il faut saisir quelque
C             chose.
C             (si la saisie doit se faire a partir d'un fichier,
C             ZC aura la valeur false apres une lecture).

```

C Fonctionnement general :

```

C  si la saisie doit se faire a partir d'un fichier, on a:
C    a - une lecture et affichage au terminal d'un message
C        d'erreur s'il y a eu un probleme.
C    b - une validation (si la lecture s'est effectuee correc-
C        tement) et affichage au terminal d'un message
C        d'erreur si ce que l'on vient de saisir ne verifie
C        pas les contraintes imposees.
C    c - retour au programme appelant.
C
C  si la lecture doit se faire a partir d'un terminal, on a:
C    a - affichage d'un message de demande au terminal
C    b - lecture
C    c - s'il y a eu un probleme, affichage d'un message
C        d'erreur et retour en a; sinon aller en d.
C    d - validation
C    e - si ce que l'on vient de saisir ne verifie pas les
C        contraintes imposees, affichage d'un message d'erreur
C        et retour en a;
C        sinon, retour au programme appelant.

```

```

C+++++C

```

```

C  SAISIE D'UN NOMBRE ENTIER POUVANT PRENDRE DEUX VALEURS.
C  =====

```

```

C  NB  : Nombre entier a saisir et a valider.
C  NB1 : Première valeur acceptable pour NB.
C  NB2 : Deuxième valeur acceptable pour NB.

```

```

C  -----
C  Contrainte : NB doit etre egal, soit a NB1, soit a NB2.
C  -----

```

```

SUBROUTINE SAVAI(NUMTER,NR,NW,NB,NR1,NR2,NMSGT,NMSGE1
1,NMSGF2,ZP)
IMPLICIT LOGICAL (Z)
ZC=.TRUE.
ZP=.FALSE.
DO 10 WHILE(ZC)
  IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
  CALL LFC1(NR,NR,ZL)
  IF (ZL) THEN
    CALL MSGFRP(NW,NMSGE1)
    ZP=.TRUE.
  ELSE
    IF (ZVLD2(NB,NR1,NR2)) THEN
      CALL MSGFRP(NW,NMSGF2)
      ZP=.TRUE.
    ELSE
      ZC=.FALSE.
    END IF
  END IF
  IF (NR.NE.NUMTER) ZC=.FALSE.
10 CONTINUE
IF (NR.EQ.NUMTER) ZP=.FALSE.
RETURN
END

```

+++++

```

C      SATISF D'UN NOMBRE ENTIER COMPRIS ENTRE DEUX BORNES.
C      =====

```

```

C      NB      : Nombre entier a saisir et a valider
C      NBMIN   : Valeur minimale acceptable pour NB.
C      NBMAX   : Valeur maximale acceptable pour NB.

```

```

C      NBMIN <= NBMAX

```

```

C      -----
C      Contrainte : NB doit etre superieur ou egal a NBMIN et
C                  inferieur ou egal a NBMAX.
C      -----

```

```

SUBROUTINE SAVA2(NUMTER,NR,NW,NB,NBMIN,NBMAX,NMSGT,NMSGE1
1,NMSGF2,ZP)
IMPLICIT LOGICAL (Z)
ZC=.TRUE.
ZP=.FALSE.
DO 10 WHILE(ZC)
  IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
  CALL LFC1(NR,NR,ZL)
  IF (ZL) THEN
    CALL MSGFRP(NW,NMSGE1)
    ZP=.TRUE.
  ELSE
    IF (ZVLD1(NB,NBMIN,NBMAX)) THEN
      CALL MSGFRP(NW,NMSGF2)
      ZP=.TRUE.
    ELSE
      ZC=.FALSE.
    END IF
  END IF
  IF (NR.NE.NUMTER) ZC=.FALSE.
10 CONTINUE
IF (NR.EQ.NUMTER) ZP=.FALSE.
RETURN
END

```



```

      END IF
      IF (NR.NE.NUMTER) ZC=.FALSE.
10    CONTINUE
      IF (NR.EQ.NUMTER) ZP=.FALSE.
      RETURN
      END

```

+++++

```

C      SAISIE D'UN NOMBRE ENTIER POUVANT PRENDRE TROIS VALEURS.
C      =====

```

```

C      NB : Nombre entier a saisir et a valider.
C      NB1 : Première valeur acceptable pour NB.
C      NB2 : Deuxième valeur acceptable pour NB.
C      NB3 : Troisième valeur acceptable pour NB.

```

```

C      -----
C      Contrainte : NB doit être égal, soit a NB1, soit a NB2, soit
C                   a NB3.
C      -----

```

```

SUBROUTINE SAVA3(NUMTER,NR,NW,NB,NB1,NB2,NB3,NMSGT,NMSCE1
1,NMSGF2,ZP)
      IMPLICIT LOGICAL (Z)
      ZC=.TRUE.
      ZP=.FALSE.
      DO 10 WHILE(ZC)
        IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
        CALL LFC1(NR,NB,ZL)
        IF (ZL) THEN
          CALL MSGERR(NW,NMSCE1)
          ZP=.TRUE.
        ELSE
          IF (ZVLD3(NB,NB1,NB2,NB3)) THEN
            CALL MSGERR(NW,NMSCE2)
            ZP=.TRUE.
          ELSE
            ZC=.FALSE.
          END IF
        END IF
      END IF
      IF (NR.NE.NUMTER) ZC=.FALSE.
10    CONTINUE
      IF (NR.EQ.NUMTER) ZP=.FALSE.
      RETURN
      END

```

+++++

```

C      SAISIE D'UN NOMBRE ENTIER SUPERIEUR OU EGAL A UNE CERTAINE
C      BORNE.
C      =====

```

```

C      NB : Nombre entier a saisir et a valider.
C      NBMIN : Valeur minimale acceptable pour NB.

```

```

C      -----
C      Contrainte : NB doit être supérieur ou égal a NBMIN.
C      -----

```

```

SUBROUTINE SAVA4(NUMTER,NR,NW,NB,NBMIN,NMSGT
1,NMSGF1,NMSCE2,ZP)

```

```

IMPLICIT LOGICAL (Z)
ZC=.TRUE.
ZP=.FALSE.
DO 10 WHILE(ZC)
  IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
  CALL LFC1(NR,NB,ZL)
  IF (ZL) THEN
    CALL MSGERR(NW,NMSGE1)
    ZP=.TRUE.
  ELSE
    IF (ZVDD4(NB,NBMIN)) THEN
      CALL MSGERR(NW,NMSGE2)
      ZP=.TRUE.
    ELSE
      ZC=.FALSE.
    END IF
  END IF
  IF (NR.NE.NUMTER) ZC=.FALSE.
10 CONTINUE
IF (NR.EQ.NUMTER) ZP=.FALSE.
RETURN
END

```

+++++

```

C      SAISIE D'UN NOM DE FICHIER DE DONNEES.
C      =====

```

```

C      FN : nom du fichier a saisir et a valider.
C      -----

```

```

C      Contrainte : un fichier de nom FN doit exister et etre non
C                   vide.
C      -----

```

```

SUBROUTINE SAVB5(NUMTER,NR,NW,FN,NMSGT,NMSGE1,NMSGE2,ZP)
IMPLICIT LOGICAL (Z)
CHARACTER*(*) FN
ZC=.TRUE.
ZP=.FALSE.
DO 10 WHILE(ZC)
  IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
  CALL LFC5(NR,FN,ZL)
  IF (ZL) THEN
    CALL MSGERR(NW,NMSGE1)
    ZP=.TRUE.
  ELSE
    IF (ZVLD5(FN)) THEN
      CALL MSGERR(NW,NMSGE2)
      ZP=.TRUE.
    ELSE
      ZC=.FALSE.
    END IF
  END IF
  IF (NR.NE.NUMTER) ZC=.FALSE.
10 CONTINUE
IF (NR.EQ.NUMTER) ZP=.FALSE.
RETURN
END

```

+++++


```

C      SATSTE D'UN NOMBRE REEL STRICTEMENT COMPRIS ENTRE DEUX BÔR-
C      NES.
C      =====

C      RB      : Nombre reel a saisir et a valider.
C      RBINF   : Borne inferieure stricte pour RB.
C      RBSUP   : Borne superieure stricte pour RB.
C      EPS     : Precision sur la borne.

C      RBINF + 2 EPS <= RBSUP

C      -----
C      Contrainte : RB doit etre strictement superieur a RBINF a EPS
C                  pres et il doit etre strictement inferieur a
C                  RBSUP a EPS pres.
C                  i.e. RB doit etre superieur ou egal a RBINF+EPS
C                  et inferieur ou egal a RBSUP-EPS.
C      -----

SUBROUTINE SAVAB(NUMTER,NR,NW,RB,RBINF,RBSUP,EPS,NMSGT
1,NMSG1,NMSG2,ZP)
IMPLICIT LOGICAL (Z)
IMPLICIT DOUBLE PRECISION (A-H,O-Y)
ZC=.TRUE.
ZP=.FALSE.
DO 10 WHILE(ZC)
  IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
  CALL LFC3(NR,RB,ZL)
  IF (ZL) THEN
    CALL MSGERR(NW,NMSG1)
    ZP=.TRUE.
  ELSE
    IF (ZV.D5(RB,RBINF,RBSUP,EPS)) THEN
      CALL MSGERR(NW,NMSG2)
      ZP=.TRUE.
    ELSE
      ZC=.FALSE.
    END IF
  END IF
  IF (NR.NE.NUMTER) ZC=.FALSE.
10 CONTINUE
IF (NR.EQ.NUMTER) ZP=.FALSE.
RETURN
END

```

C+++++

```

C      SATSTE D'UN NOMBRE REEL COMPRIS ENTRE DEUX BÔRNES.
C      =====

```

```

C      RB      : Nombre reel a saisir et a valider.
C      RBMIN   : Valeur minimale acceptable pour RB.
C      RBMAX   : Valeur maximale acceptable pour RB.

```

```

C      RBMIN <= RBMAX

```

```

C      -----
C      Contrainte : RB doit etre superieur ou egal a RBMIN et infe-
C                  rieur ou egal a RBMAX.

```

C

```

-----
SUBROUTINE SAVAT(NUMTER,NR,NW,PB,RBMIN,RBMAX,NMSGT,NMSGE1
1,NMSGE2,ZP)
IMPLICIT LOGICAL (Z)
IMPLICIT DOUBLE PRECISION (A-H,O-Y)
ZC=.TRUE.
ZP=.FALSE.
DO 10 WHILE(ZC)
  IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
  CALL DEC3(NR,RB,ZL)
  IF (ZL) THEN
    CALL MSGFRP(NW,NMSGE1)
    ZP=.TRUE.
  ELSE
    IF (ZVLD7(PB,RBMIN,RBMAX)) THEN
      CALL MSGFRP(NW,NMSGE2)
      ZP=.TRUE.
    ELSE
      ZC=.FALSE.
    END IF
  END IF
  IF (NR.NE.NUMTER) ZC=.FALSE.
10 CONTINUE
  IF (NR.EQ.NUMTER) ZP=.FALSE.
  RETURN
END

```

C+++++

C SAISTE D'UN VECTEUR D'ENTIERS DISTINCTS ET COMPRIS ENTRE DEUX
C bornes.
C =====

C NVR : Identificateur du vecteur d'entiers a saisir et a
C valider.
C NB : Nombre de composantes a saisir et a valider.
C IDIM : dimension du vecteur.
C NBMIN : Valeur minimale acceptable pour les composantes.
C NBMAX : Valeur maximale acceptable pour les composantes.

C 1 <= NR <= IDIM
C NBMIN <= NBMAX

C -----
C Contrainte : NVB(I), pour I allant de 1 a NB, doit etre supe-
C rieur ou egal a NBMIN et inferieur ou egal a
C NBMAX.
C Pour I allant de 1 a NB et pour J allant de 1 a
C NB, J different de I, NVB(I) doit etre different
C de NVB(J).
C -----

```

SUBROUTINE SAVAB(NUMTER,NR,NW,NVR,NB,IDIM,NBMIN,NBMAX,NMSGT
1,NMSGE1,NMSGE2,NMSGE3,ZP)
IMPLICIT LOGICAL (Z)
DIMENSION NVR(IDIM)
ZC=.TRUE.
ZP=.FALSE.
DO 10 WHILE(ZC)

```



```

      IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
      CALL LEC2(NR,NVB,NB,IDIM,ZL)
      IF (ZL) THEN
        CALL MSGERR(NW,NMSGE1)
        ZP=.TRUE.
      ELSE
        IF (ZVLD8(NVB,NB,IDIM,NBMIN,NBMAX)) THEN
          CALL MSGERR(NW,NMSGE2)
          ZP=.TRUE.
        ELSE
          IF (ZVLD9(NVB,NB,IDIM)) THEN
            CALL MSGERR(NW,NMSGE3)
            ZP=.TRUE.
          ELSE
            ZC=.FALSE.
          END IF
        END IF
      END IF
      IF (NR.NE.NUMTER) ZC=.FALSE.
10  CONTINUE
      IF (NR.EQ.NUMTER) ZP=.FALSE.
      RETURN
      END

```

+++++

```

C      SATSTE D'UN VECTEUR D'ENTIERS COMPRIS ENTRE DEUX BORNES.
C      =====

```

```

C      NVB   : Identificateur du vecteur d'entiers a saisir et a
C              valider.
C      NB    : Nombre de composantes a saisir et a valider.
C      IDIM  : dimension du vecteur.
C      NBMIN : Valeur minimale acceptable pour les composantes.
C      NBMAX : Valeur maximale acceptable pour les composantes.

```

```

C      1 <= NB <= IDIM
C      NBMIN <= NBMAX

```

```

C      -----
C      Contrainte : NVB(I), pour I allant de 1 a NB, doit etre supe-
C                   rieur ou egal a NBMIN et inferieur ou egal a
C                   NBMAX.
C      -----

```

```

SUBROUTINE SAVA9(NUMTER,NR,NW,NVB,NB,IDIM,NBMIN,NBMAX
1,NMSGT,NMSGE1,NMSGE2,ZP)
IMPLICIT LOGICAL (Z)
DIMENSION NVB(IDIM)
ZC=.TRUE.
ZP=.FALSE.
DO 10 WHILE(ZC)
  IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
  CALL LEC2(NR,NVB,NB,IDIM,ZL)
  IF (ZL) THEN
    CALL MSGERR(NW,NMSGE1)
    ZP=.TRUE.
  ELSE
    IF (ZVLD8(NVB,NB,IDIM,NBMIN,NBMAX)) THEN
      CALL MSGERR(NW,NMSGE2)

```

```

        ZP=.TRUE.
      ELSE
        ZC=.FALSE.
      END IF
    END IF
    IF (NR.NE.NUMTER) ZC=.FALSE.
10 CONTINUE
    IF (NR.EQ.NUMTER) ZP=.FALSE.
    RETURN
  END

```

+++++

```

C      SATISTE D'UN VECTEUR D'ENTIERES DE SOMME DONNEE ET COMPTES
C      ENTRE DEUX BORNES DONNEES.
C      =====

```

```

C      NVR      : Identificateur du vecteur d'entiers a saisir et a
C                valider.
C      NB       : Nombre de composantes a saisir et a valider.
C      IDIM     : dimension du vecteur.
C      NBMIN    : Valeur minimale acceptable pour les composantes.
C      NBMAX    : Valeur maximale acceptable pour les composantes.
C      NSOM     : Valeur demandee pour la somme des NB premieres
C                composantes.

```

```

C      1 <= NR <= IDIM
C      NBMIN <= NBMAX

```

```

C      -----
C      Contrainte : NVR(I), pour I allant de 1 a NB, doit etre supe-
C                  rieur ou egal a NBMIN et inferieur ou egal a
C                  NBMAX.
C                  La somme des NVR(I) pour I allant de 1 a NB doit
C                  valoir NSOM.
C      -----

```

```

SUBROUTINE SAVA10(NUMTER,NR,NW,NVB,NR,IDIM,NBMIN,NBMAX,NSOM
1,NMSGT,NMSCE1,NMSCE2,NMSCE3,ZP)

```

```

IMPLICIT LOGICAL (Z)

```

```

DIMENSION NVR(IDIM)

```

```

ZC=.TRUE.

```

```

ZP=.FALSE.

```

```

DO 10 WHILE(ZC)

```

```

  IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)

```

```

  CALL LFC2(NR,NVR,NR,IDIM,ZL)

```

```

  IF (ZL) THEN

```

```

    CALL MSGERR(NW,NMSCE1)

```

```

    ZP=.TRUE.

```

```

  ELSE

```

```

    IF (ZVLD1(NVR,NB,IDIM,NBMIN,NBMAX)) THEN

```

```

      CALL MSGERR(NW,NMSCE2)

```

```

      ZP=.TRUE.

```

```

    ELSE

```

```

      IF (ZVLD10(NVR,NR,IDIM,NSOM)) THEN

```

```

        CALL MSGERR(NW,NMSCE3)

```

```

        ZP=.TRUE.

```

```

      ELSE

```

```

        ZC=.FALSE.

```

```

      END IF

```



```

      END IF
      END IF
      IF (NR.NE.NUMTER) ZC=.FALSE.
10    CONTINUE
      IF (NR.EQ.NUMTER) ZP=.FALSE.
      RETURN
      END
C+++++
C
C   SAISIE D'UN VECTEUR D'ENTIERES COMPRENS ENTRE DEUX BORNES
C   DONNEES ET VERIFIANT LA CONDITION DE REGROUPEMENT.
C   =====
C
C   NVR   : Identificateur du vecteur d'entiers a saisir et a
C           valider.
C   NB    : Nombre de composantes a saisir et a valider.
C   TDTM  : dimension du vecteur.
C   NBMIN : Valeur minimale acceptable pour les composantes.
C   NBMAX : Valeur maximale acceptable pour les composantes.
C
C   1 <= NB <= TDTM
C   NBMIN <= NBMAX
C   1 <= NVB(I) <= NB pour I allant de 1 a NB.
C
C   -----
C   Contrainte : NVB(I), pour I allant de 1 a NB, doit etre
C                 superieur ou egal a NBMIN et inferieur ou egal a
C                 NBMAX.
C                 Pour chaque composante I de ce vecteur, I allant
C                 de 1 a NB, si cette composante vaut K, alors la
C                 Kieme composante vaut K.
C   -----
C
SUBROUTINE SAVAIL(NUMTER,NP,NW,NVB,NR,TDIM,NBMIN,NBMAX,NMSGT
1,NMSGF1,NMSGF2,NMSGF3,ZP)
IMPLICIT LOGICAL (Z)
DIMENSION NVR(TDIM)
ZC=.TRUE.
ZP=.FALSE.
DO 10 WHILE(ZC)
  IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
  CALL LFC2(NR,NVB,NR,TDIM,ZL)
  IF (ZL) THEN
    CALL MSGF1(NW,NMSGF1)
    ZP=.TRUE.
  ELSE
    IF (ZVIDR(NVR,NR,TDIM,NBMIN,NBMAX)) THEN
      CALL MSGF2(NW,NMSGF2)
      ZP=.TRUE.
    ELSE
      IF (ZVID1(NVR,NR,TDIM)) THEN
        CALL MSGF3(NW,NMSGF3)
        ZP=.TRUE.
      ELSE
        ZC=.FALSE.
      END IF
    END IF
  END IF
END IF
END IF
IF (NR.NE.NUMTER) ZC=.FALSE.
10 CONTINUE

```

```

      IF (NR.EQ.NUMTER) ZP=.FALSE.
      RETURN
      END

```

+++++

```

C      SAISIE D'UN NOM DE FICHIER DE RESULTATS.
C      =====

```

```

C      FN : Nom du fichier a saisir.
C      -----

```

```

C      Contrainte : neant
C      -----

```

```

      SUBROUTINE SAVAI2(NUMTER,NR,NW,FN,NMSGT,NMSGF1,ZP)

```

```

      IMPLICIT LOGICAL (Z)

```

```

      CHARACTER*(*) FN

```

```

      ZC=.TRUE.

```

```

      ZP=.FALSE.

```

```

      DO 10 WHILE(ZC)

```

```

         IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)

```

```

         CALL DECS(NR,FN,ZC)

```

```

         IF (ZC) THEN

```

```

            CALL MSGERR(NW,NMSGF1)

```

```

            ZP=.TRUE.

```

```

         ELSE

```

```

            ZC=.FALSE.

```

```

         END IF

```

```

         IF (NR.NE.NUMTER) ZC=.FALSE.

```

```

10      CONTINUE

```

```

      IF (NR.EQ.NUMTER) ZP=.FALSE.

```

```

      RETURN

```

```

      END

```

+++++

```

C      SAISIE D'UN VECTEUR D'ENTIERES CROISSANTS ET COMPTES ENTRE
C      DEUX BORNES DONNEES.
C      =====

```

```

C      NVR   : Identificateur du vecteur d'entiers a saisir et a
C              valider.

```

```

C      NB    : Nombre de composantes a saisir et a valider.

```

```

C      TDM   : dimension du vecteur.

```

```

C      NBMIN : Valeur minimale acceptable pour les composantes.

```

```

C      NBMAX : Valeur maximale acceptable pour les composantes.

```

```

C      1 <= NR <= TDM

```

```

C      NBMIN <= NBMAX

```

```

C      -----
C      Contrainte : NVR(I), pour I allant de 1 a NB, doit etre supe-
C                    rieur ou egal a NBMIN et inferieur ou egal a
C                    NBMAX.

```

```

C                    NVR(1) < ... < NVR(NB) si NB > 1.
C      -----

```

```

      SUBROUTINE SAVAI3(NUMTER,NR,NW,NVB,NB,TDM,NBMIN,NBMAX

```

```

      1,NMSGT,NMSGF1,NMSGF2,NMSGF3,ZP)

```

```

      IMPLICIT LOGICAL (Z)

```

```

      DIMENSION NVR(TDM)

```



```

7C=.TRUE.
7P=.FALSE.
DO 10 WHILE(7C)
  IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
  CALL LFC2(NR,NVB,NR,TDIM,ZL)
  IF (ZL) THEN
    CALL MSGERR(NW,NMSG1)
    7P=.TRUE.
  ELSE
    IF (7VLD1(NVB,NR,TDIM,NBMIN,NBMAX)) THEN
      CALL MSGERR(NW,NMSG2)
      7P=.TRUE.
    ELSE
      IF (7VLD12(NVB,NR,TDIM)) THEN
        CALL MSGERR(NW,NMSG3)
        7P=.TRUE.
      ELSE
        7C=.FALSE.
      END IF
    END IF
  END IF
  IF (NR.NE.NUMTER) 7C=.FALSE.
CONTINUE
IF (NR.EQ.NUMTER) 7P=.FALSE.
RETURN
END

```

10

+++++

```

C      SATSIE D'UN VECTEUR DE REELS DE SOMME DONNEE ET STRICTEMENT
C      COMPRIS ENTRE DEUX BORNES DONNEES.
C      =====

```

```

C      RVB      : Identificateur du vecteur de reels a saisir et a
C                  valider.
C      NB       : Nombre de composantes a tester.
C      TDIM     : dimension du vecteur.
C      RBINF    : Borne inferieure stricte pour RB.
C      RBSUP    : Borne superieure stricte pour RB.
C      EPS1     : Precision sur les bornes.
C      RSOM     : Valeur demandee pour la somme des NB premieres com-
C                  posantes.
C      EPS2     : Precision sur la somme.

```

```

C      1 <= NB <= TDIM
C      RBINF + 2 EPS <= RBSUP

```

```

C      -----
C      Contrainte : Pour chaque composante I de ce vecteur, I allant
C                  de 1 a NB, RVB(I) doit etre strictement compris
C                  entre RBINF et RBSUP.
C                  i.e. RVB(I) doit etre superieur ou egal a RBINF
C                  +EPS1 et strictement inferieur a RBSUP-EPS1.
C                  La difference en valeur absolue entre la somme
C                  des NB premieres composantes de RVB et RSOM doit
C                  etre inferieure ou egale a EPS2.
C      -----

```

SUBROUTINE SAV14(NUMTER,NR,NW,RVB,NR,TDIM,RRINF,RBSUP,EPS1

```

1,RSOM,EDS2,NMSGT,NMSGF1,NMSGF2,NMSGF3,ZD)
IMPLICIT LOGICAL (Z)
IMPLICIT DOUBLE PRECISION (A-H,O-Y)
DIMENSION RVR(TDTM)
ZC=.TRUE.
ZP=.FALSE.
DO 10 WHILE(ZC)
  IF (NR.EQ.NUMTER) CALL MSGT(NW,NMSGT)
  CALL DECA(NR,RVR,NR,TDTM,ZI)
  IF (ZI) THEN
    CALL MSGFRP(NW,NMSGF1)
    ZP=.TRUE.
  ELSE
    IF (ZVLD13(RVR,NR,TDTM,RRTIME,RPSUP,EDS1)) THEN
      CALL MSGFRP(NW,NMSGF2)
      ZP=.TRUE.
    ELSE
      IF (ZVLD14(RVR,NR,TDTM,RSOM,EDS2)) THEN
        CALL MSGFRP(NW,NMSGF3)
        ZP=.TRUE.
      ELSE
        ZC=.FALSE.
      END IF
    END IF
  END IF
END IF
IF (NR.NE.NUMTER) ZC=.FALSE.
CONTINUE
IF (NR.EQ.NUMTER) ZP=.FALSE.
RETURN
END

```



```

C*****C
C*                                           *C
C*          BLOC DES FONCTIONS DE VALIDATION SIMPLE          *C
C*                                           *C
C*****C

```

```

C      Chaque fonction de ce bloc retourne la valeur :
C      TRUE, si la contrainte a verifier n'est pas respectee
C      FALSE, sinon.

```

```

C+++++

```

```

C      VALIDATION D'UN NOMBRE ENTIER COMPREIS ENTRE DEUX BORNES.
C      =====

```

```

C      NB      : Nombre entier a tester
C      NBMIN   : Valeur minimale acceptable pour NB
C      NBMAX   : Valeur maximale acceptable pour NB

```

```

C      NBMIN <= NBMAX

```

```

C      -----
C      Contrainte : NB doit etre superieur ou egal a NBMIN et
C                  interieur ou egal a NBMAX.
C      -----

```

```

LOGICAL FUNCTION ZVLD1(NB,NBMIN,NBMAX)
IF ((NB.LT.NBMIN).OR.(NB.GT.NBMAX)) THEN
  ZVLD1=.TRUE.
ELSE
  ZVLD1=.FALSE.
END IF
RETURN
END

```

```

C+++++

```

```

C      VALIDATION D'UN NOMBRE ENTIER POUVANT PRENDRE DEUX VALEURS.
C      =====

```

```

C      NB      : Nombre entier a tester
C      NB1     : Premiere valeur acceptable pour NB
C      NB2     : Deuxieme valeur acceptable pour NB

```

```

C      -----
C      Contrainte : NB doit etre egal, soit a NB1, soit a NB2.
C      -----

```

```

LOGICAL FUNCTION ZVLD2(NB,NB1,NB2)
IF ((NB.EQ.NB1).AND.(NB.EQ.NB2)) THEN
  ZVLD2=.TRUE.
ELSE
  ZVLD2=.FALSE.
END IF
RETURN
END

```

```

C+++++

```

```

C      VALIDATION D'UN NOMBRE ENTIER POUVANT PRENDRE TROIS VALEURS.
C      =====

```

```

C      NB      : Nombre entier a tester
C      NB1     : Premiere valeur acceptable pour NB
C      NB2     : Deuxieme valeur acceptable pour NB
C      NB3     : Troisieme valeur acceptable pour NB
C      -----
C      Contrainte : NB doit etre egal, soit a NB1, soit a NB2, soit
C                   a NB3.
C      -----

```

```

LOGICAL FUNCTION ZVLD3(NB,NB1,NB2,NB3)
IF ((NB.NE.NB1).AND.(NB.NE.NB2).AND.(NB.NE.NB3)) THEN
  ZVLD3=.TRUE.
ELSE
  ZVLD3=.FALSE.
END IF
RETURN
END

```

+++++

```

C      VALIDATION D'UN NOMBRE ENTIER SUPERIEUR OU EGAL A UNE
C      CERTAINE BORNES.
C      =====

```

```

C      NB      : Nombre entier a tester
C      NBMIN    : Valeur minimale acceptable pour NB
C      -----
C      Contrainte : NB doit etre superieur ou egal a NBMIN
C      -----

```

```

LOGICAL FUNCTION ZVLD4(NB,NBMIN)
IF (NB.GT.NBMIN) THEN
  ZVLD4=.TRUE.
ELSE
  ZVLD4=.FALSE.
END IF
RETURN
END

```

+++++

```

C      VALIDATION D'UN NOM DE FICHIER DE DONNEES.
C      =====

```

```

C      FN      : Nom du fichier
C      -----
C      Contrainte : Le fichier de nom FN doit exister et etre non
C                   vide.
C      -----

```

```

LOGICAL FUNCTION ZVLD5(FN)
CHARACTER*(*) FN
OPEN(UNIT=40,FILE=FN,DEVICE='DSK',ERR=10)
READ(40,*,ERR=10,END=10)
CLOSE(40)
ZVLD5=.FALSE.
RETURN
ZVLD5=.TRUE.
RETURN

```


F4D

C+++++

```
C      VALIDATION D'UN NOMBRE REEL STRICTEMENT COMPRIS ENTRE DEUX
C      BORNES.
C      =====
```

```
C      RB      : Nombre reel a tester
C      RBINF   : Borne inferieure stricte pour RB
C      RBSUP   : Borne superieure stricte pour RB
C      EPS     : Precision sur la borne
```

```
C      RBSUP >= RBINF + 2 EPS
```

```
C      -----
C      Contrainte : RB doit etre strictement superieur a RBINF a EPS
C                  pres et il doit etre strictement inferieur a
C                  RBSUP a EPS pres.
C                  i.e. RB doit etre superieur ou egal a RBINF+EPS
C                  et inferieur ou egal a RBSUP-EPS.
C      -----
```

```
LOGICAL FUNCTION ZVLD6(RB,RBINF,RBSUP,EPS)
IMPLICIT DOUBLE PRECISION (A-H,O-Y)
IF ((RB.LT.(RBINF+EPS)).OR.(RB.GT.(RBSUP-EPS))) THEN
    ZVLD6=.TRUE.
ELSE
    ZVLD6=.FALSE.
END IF
RETURN
END
```

C+++++

```
C      VALIDATION D'UN NOMBRE REEL COMPRIS ENTRE DEUX BORNES.
C      =====
```

```
C      RB      : Nombre reel a tester
C      RBMIN   : Valeur minimale acceptable pour RB
C      RBMAX   : Valeur maximale acceptable pour RB
```

```
C      RBMIN <= RBMAX
```

```
C      -----
C      Contrainte : RB doit etre superieur ou egal a RBMIN et
C                  inferieur ou egal a RBMAX
C      -----
```

```
LOGICAL FUNCTION ZVLD7(RB,RBMIN,RBMAX)
IMPLICIT DOUBLE PRECISION (A-H,O-Y)
IF ((RB.LT.RBMIN).OR.(RB.GT.RBMAX)) THEN
    ZVLD7=.TRUE.
ELSE
    ZVLD7=.FALSE.
END IF
RETURN
END
```

C+++++

```

C      VALIDATION D'UN VECTEUR D'ENTIERS COMPRIS ENTRE DEUX BORNES.
C      =====

```

```

C      NVR      : Identificateur du vecteur d'entiers a tester
C      NB       : Nombre de composantes a tester
C      TDTM     : Dimension du vecteur
C      NBMIN    : Valeur minimale acceptable pour les composantes
C      NBMAX    : Valeur maximale acceptable pour les composantes

```

```

C      1 <= NB <= TDTM
C      NBMIN <= NBMAX

```

```

C      -----
C      Contrainte : NVR(I), pour I allant de 1 a NB, doit etre
C                   superieur ou egal a NBMIN et inferieur ou egal
C                   a NBMAX.
C      -----

```

```

C      LOGICAL FUNCTION ZVLD8(NVR,NB,TDIM,NBMIN,NBMAX)
C      DIMENSION NVR(TDIM)
C      DO 10 I=1,NB
C         IF ((NVR(I).LT.NBMIN).OR.(NVR(I).GT.NBMAX)) THEN
C            ZVLD8=.TRUE.
C            RETURN
C         END IF
10    CONTINUE
C      ZVLD8=.FALSE.
C      RETURN
C      END

```

```

C+++++

```

```

C      VALIDATION D'UN VECTEUR D'ENTIERS DISTINCTS.
C      =====

```

```

C      NVR      : Identificateur du vecteur d'entiers a tester
C      NB       : Nombre de composantes a tester
C      TDTM     : Dimension du vecteur

```

```

C      1 <= NB <= TDTM

```

```

C      -----
C      Contrainte : Pour I allant de 1 a NB et pour J allant de 1 a
C                   NB, J différent de I, NVR(I) doit etre différent
C                   de NVR(J).
C      -----

```

```

C      LOGICAL FUNCTION ZVLD9(NVR,NB,TDIM)
C      DIMENSION NVR(TDIM)
C      IF (NB.EQ.1) THEN
C         ZVLD9=.FALSE.
C      ELSE
C         DO 10 I=2,NB
C            DO 20 J=1,I-1
C               IF (NVR(J).EQ.NVR(I)) THEN
C                  ZVLD9=.TRUE.
C                  RETURN
C               END IF
20          CONTINUE
10    CONTINUE

```



```

10      CONTINUE
        ZVLD0=.FALSE.
      END IF
      RETURN
    END

```

+++++

```

C      VALIDATION D'UN VECTEUR D'ENTIERES DE SOMME DONNEE.
C      =====

```

```

C      NVR : Identificateur du vecteur d'entiers a tester
C      NB  : Nombre de composantes a tester
C      TDIM : Dimension du vecteur
C      NSOM : Valeur demandee pour la somme des composantes a tester

```

```

C      1 <= NB <= TDIM

```

```

C      -----
C      Contrainte : La somme des NB premieres composantes de NVR
C                   doit valoir NSOM.
C      -----

```

```

      LOGICAL FUNCTION ZVLD10(NVR,NB,TDIM,NSOM)
      DIMENSION NVR(TDIM)
      NSOMA=0
      DO 10 I=1,NB
        NSOMA=NSOMA+NVR(I)
10     CONTINUE
      IF (NSOMA.NE.NSOM) THEN
        ZVLD10=.TRUE.
      ELSE
        ZVLD10=.FALSE.
      END IF
      RETURN
    END

```

+++++

```

C      VALIDATION D'UN VECTEUR D'ENTIERES VERIFIANT LA CONDITION DE
C      REGROUPEMENT.
C      =====

```

```

C      NVR : Identificateur du vecteur d'entiers a tester
C      NB  : Nombre de composantes a tester
C      TDIM : Dimension du vecteur

```

```

C      1 <= NB <= TDIM
C      1 <= NVR(I) <= NB pour I allant de 1 a NB

```

```

C      -----
C      Contrainte : Pour chaque composante I de ce vecteur, I allant
C                   de 1 a NB, si cette composante vaut K, alors la
C                   Kieme composante vaut K.
C      -----

```

```

      LOGICAL FUNCTION ZVLD11(NVR,NB,TDIM)
      DIMENSION NVR(TDIM)
      DO 10 I=1,NB
        IF (NVR(NVR(I)).NE.NVR(I)) THEN

```

```

        ZVID11=.TRUE.
        RETURN
    END IF
10    CONTINUE
    ZVID11=.FALSE.
    RETURN
END

```

+++++

```

C      VALIDATION D'UN VECTEUR D'ENTIERS STRICTEMENT CROISSANTS.
C      =====

```

```

C      NVR   : Identificateur du vecteur d'entiers a tester
C      NB    : Nombre de composantes a tester
C      TDM   : Dimension du vecteur

```

```

C      1 <= NB <= TDM

```

```

C      -----
C      Contrainte : NVR(1) < ... < NVR(NB) si NB > 1
C      -----

```

```

LOGICAL FUNCTION ZVID12(NVR,NB,TDM)
DIMENSION NVR(TDM)
IF (NB.EQ.1) THEN
    ZVID12=.FALSE.
ELSE
    DO 10 I=2,NB
        IF (NVR(I-1).GE.NVR(I)) THEN
            ZVID12=.TRUE.
            RETURN
        END IF
10    CONTINUE
    ZVID12=.FALSE.
END IF
RETURN
END

```

+++++

```

C      VALIDATION D'UN VECTEUR DE REELS STRICTEMENT COMPRTS ENTRE
C      DEUX BORNES DONNEES.
C      =====

```

```

C      RVR    : Identificateur du vecteur de reels a tester
C      NB     : Nombre de composantes a tester
C      TDM    : Dimension du vecteur
C      RBINF  : Borne inferieure stricte pour RB
C      RBSUP  : Borne superieure stricte pour RB
C      EPS    : Precision sur les bornes

```

```

C      1 <= NB <= TDM
C      RBINF + 2 EPS <= RBSUP

```

```

C      -----
C      Contrainte : Pour chaque composante I de ce vecteur, I allant
C                  de 1 a NB, RVR(I) doit etre strictement compris
C                  entre RBINF et RBSUP.
C                  i.e. RVR(I) doit etre superieur ou egal a

```


RBINF+EPS et strictement inférieur a PBSUP-EPS.

```
-----
LOGICAL FUNCTION ZVLD13(RVB,NB,IDIM,RBINF,PBSUP,EPS)
IMPLICIT DOUBLE PRECISION (A-H,O-Y)
DIMENSION RVB(IDIM)
DO 10 I=1,NB
  IF ((RVB(I).LT.(RBINF+EPS)).OR.
1  (RVB(I).GT.(PBSUP-EPS))) THEN
    ZVLD13=.TRUE.
    RETURN
  END IF
10 CONTINUE
ZVLD13=.FALSE.
RETURN
END
```

+++++
VALIDATION D'UN VECTEUR DE REELS DE SOMME DONNEE.
=====

RVB : Identificateur du vecteur a tester
NB : Nombre de composantes a tester
IDIM : Dimension du vecteur
PSOM : Valeur normale de la somme des NB premieres
composantes de RVB
EPS : Precision sur la somme

1 <= NB <= IDIM

Contrainte : La difference en valeur absolue entre la somme
des NB premieres composantes de RVB et PSOM doit
etre inferieure ou egale a EPS.

```
LOGICAL FUNCTION ZVLD14(RVB,NB,IDIM,PSOM,EPS)
IMPLICIT DOUBLE PRECISION (A-H,O-Y)
DIMENSION RVB(IDIM)
PSOMA=0.
DO 10 I=1,NB
  PSOMA=PSOMA+RVB(I)
10 CONTINUE
IF (ABS(PSOMA-PSOM).GT.EPS) THEN
  ZVLD14=.TRUE.
ELSE
  ZVLD14=.FALSE.
END IF
RETURN
END
```

+++++

```

C*****
C*
C*          BLOC DES ROUTINES DE LECTURE.
C*
C*****

```

```

C      Pour chaque routine de ce bloc :
C      NR est un numero d'unité pour la lecture, NR = 5 ou
C      15 <= NR <= 60.
C      ZR est un indicateur d'existence d'erreur.
C      = true s'il y a eu une erreur lors de la lecture
C      = false sinon.

```

```

C+++++

```

```

C      LECTURE D'UN NOMBRE ENTIER.
C      =====

```

```

C      NB : Nombre entier lu
C      -----

```

```

C      SUBROUTINE LEC1(NR,NB,ZR)
C      IMPLICIT LOGICAL (Z)
C      ZR=.FALSE.
C      READ(NR,*,ERR=10,END=10)NB
C      RETURN
10  ZR=.TRUE.
C      RETURN
C      END

```

```

C+++++

```

```

C      LECTURE D'UN VECTEUR D'ENTIERS.
C      =====

```

```

C      MVR : Vecteur d'entiers lu
C      NB  : Nombre d'elements a lire
C      IDIM : Dimension du vecteur
C      -----

```

```

C      Les composantes de ce vecteur se trouvent sur une meme ligne
C      et sont separees par un nombre quelconque de caracteres ^ ou
C      par une virgule.
C      -----

```

```

C      SUBROUTINE LEC2(NR,NVB,NB,IDIM,ZR)
C      IMPLICIT LOGICAL (Z)
C      DIMENSION MVR(IDIM)
C      ZR=.FALSE.
C      READ(NR,*,ERR=10,END=10)(NVB(I),I=1,NB)
C      RETURN
10  ZR=.TRUE.
C      RETURN
C      END

```

```

C+++++

```

```

C      LECTURE D'UN NOMBRE REEL.
C      =====

```

```

C      RB : Nombre reel lu

```



```

-----
SUBROUTINE LFC3(NR,RP,7R)
IMPLICIT LOGICAL (7)
DOUBLE PRECISION RP
ZR=.FALSE.
READ(NR,*,FPR=10,END=10)PR
RETURN
ZR=.TRUE.
RETURN
END

```

10

```

+++++
C LECTURE D'UN VECTEUR DE REFLS.
C =====

```

```

C RVB : vecteur reel lu
C NR : nombre d'elements a lire
C TOTM : dimension du vecteur
C -----

```

```

C Les composantes de ce vecteur se trouvent sur une meme ligne
C et sont separees par un nombre quelconque de caracteres ^ ou
C par une virgule.
C -----

```

```

SUBROUTINE LFC4(NR,RVB,NR.IDTM,ZP)
IMPLICIT LOGICAL (7)
DOUBLE PRECISION RVB
DIMENSION RVB(TOTM)
ZR=.FALSE.
READ(NR,*,FPR=10,END=10)(RVB(I),I=1,NR)
RETURN
ZR=.TRUE.
RETURN
END

```

10

```

+++++
C LECTURE D'UN NOM DE FICHIER.
C =====

```

```

C FN : nom du fichier lu (10 caracteres)
C -----
C Les 10 premiers caracteres de la ligne representent le nom du
C fichier.
C -----

```

```

SUBROUTINE LFC5(NR,FN,7R)
IMPLICIT LOGICAL (7)
CHARACTER*(*) FN
ZR=.FALSE.
READ(NR,*(A10)',FPR=10,END=10)FN
RETURN
ZR=.TRUE.
RETURN
END

```

10

```

+++++

```

```

C      LECTURE D'UN LIBELLE DE VARIABLE.
C      =====

```

```

C      LV : Libelle lu (10 caracteres)
C      -----

```

```

C      Les 10 premiers caracteres de la ligne representent le
C      libelle de la variable.
C      -----

```

```

C      SUBROUTINE LEC6(NR,LV,ZR)
C      IMPLICIT LOGICAL (Z)
C      CHARACTER*(*) LV
C      ZR=.FALSE.
C      READ(NR,'(A10)',FPP=10,END=10)LV
C      RETURN
10    ZR=.TRUE.
C      RETURN
C      END

```

```

C+++++

```

```

C      LECTURE D'UN LIBELLE DE VARIABLE ET D'UN NOMBRE DE MODALITES.
C      =====

```

```

C      LV : Libelle lu (10 caracteres)
C      NM : Nombre de modalites lu
C      -----

```

```

C      Les 10 premiers caracteres de la ligne representent le
C      libelle de la variable, le nombre de modalites se trouve
C      apres sous la forme d'un entier.
C      -----

```

```

C      SUBROUTINE LEC7(NR,LV,NM,ZR)
C      IMPLICIT LOGICAL (Z)
C      CHARACTER*(*) LV
C      ZR=.FALSE.
C      READ(NR,'(A10,I)',FPP=10,END=10)LV,NM
C      RETURN
10    ZR=.TRUE.
C      RETURN
C      END

```

```

C+++++

```

```

C      LECTURE D'UN VECTEUR DE LIBELLES DE MODALITES.
C      =====

```

```

C      LVM : Vecteur de libelles de modalites (4 caracteres) lu
C      NM : Nombre de libelles a lire
C      TDTM : Dimension du vecteur
C      -----

```

```

C      Les quatre premiers caracteres de la ligne representent le
C      libelle de la premiere modalites: les autres suivent
C      (toujours 4 caracteres) et sont separees par un caractere ^.
C      -----

```

```

C      SUBROUTINE LEC8(NR,LVM,NM,TDIM,ZR)
C      IMPLICIT LOGICAL (Z)
C      CHARACTER*(*) LVM(TDIM)
C      ZR=.FALSE.

```


10

```
DEAD(ND,'(0(A4,X))',FRP=10,END=10)(LVH(1),T=1,MH)  
RETURN  
7R=.TRUE.  
RETURN  
END
```

C+++++


```

RETURN

7  WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU MODE DE REPARTI
   TION DES'
   WRITE(NW,*)'VARIABLES !!'
   RETURN

8  WRITE(NW,*)'MODE DE REPARTITION DES VARIABLES INVALIDE !!'
   RETURN

9  WRITE(NW,*)'PAS VALIDATION DES NUMEROS DES VARTABLES !!'
   RETURN

10 WRITE(NW,*)'ERREUR FATALE !!'
   WRITE(NW,*)'IMPOSSIBLE DE CONTINUER LA SAISIE !!'
   RETURN

11 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DES NUMEROS DES
   1 VARIABLES !!'
   RETURN

12 WRITE(NW,*)'NUMERO DE VARIABLE INVALIDE !!'
   RETURN

13 WRITE(NW,*)'NUMEROS DE VARTABLES PAS PAR ORDRE CROISSANT !!'
   RETURN

14 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOMBRE
   1 D'INDIVIDUS !!'
   RETURN

15 WRITE(NW,*)'NOMBRE D'INDIVIDUS INVALIDE !!'
   RETURN

16 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOM DU FICHER
   1 DES POIDS'
   WRITE(NW,*)'DES INDIVIDUS !!'
   RETURN

17 WRITE(NW,*)'LE FICHER DES POIDS DES INDIVIDUS EST VIDE
   1 OU INEXISTANT !!'
   RETURN

18 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DE L'INDICATEUR
   1 DE L'EXISTENCE'
   WRITE(NW,*)'D'UN FICHER DE LIBELLES !!'
   RETURN

19 WRITE(NW,*)'VALEUR INVALIDE POUR L'INDICATEUR DE L'EXISTEN
   1 CE D'UN'
   WRITE(NW,*)'FICHER DE LIBELLES !!'
   RETURN

20 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOM DU FICHER
   1 DES LIBELLES !!'
   RETURN

21 WRITE(NW,*)'FICHER DE LIBELLES VIDE OU INEXISTANT !!'
   RETURN

```

22 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DE L'INDICATEUR
1 DE L'E'GALITE'
WRITE(NW,*)'DES LIBELLES DE MODALITES POUR LES VARIABLES !!'
RETURN

23 WRITE(NW,*)'VALEUR DE L'INDICATEUR DE L'E'GALITE DES
1 LIBELLES DES'
WRITE(NW,*)'MODALITES DES VARIABLES INVALIDE !!'
RETURN

24 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DE L'INDICATEUR DE
1 L'E'GALITE DES'
WRITE(NW,*)'NOMBRES DE MODALITES POUR LES VARIABLES !!'
RETURN

25 WRITE(NW,*)'VALEUR DE L'INDICATEUR DE L'E'GALITE DES NOM
1BRES DE MODALITES'
WRITE(NW,*)'POUR LES VARIABLES INVALIDE !!'
RETURN

26 WRITE(NW,*)'PAS VALIDATION DU OU DES NOMBRES DE MODALITES !!'
RETURN

27 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOMBRE DE MODALIT
ITES !!'
RETURN

28 WRITE(NW,*)'NOMBRE DE MODALITES INVALIDE !!'
RETURN

29 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DES NOMBRES DE
1MODALITES !!'
RETURN

30 WRITE(NW,*)'NOMBRE DE MODALITES INVALIDE !!'
RETURN

31 WRITE(NW,*)'PAS VALIDATION DES NOMBRES DE MODALITES !!'
RETURN

32 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOM DU FICHER
1 DES CODAGES CONDENSES !!'
RETURN

33 WRITE(NW,*)'FICHER DES CODAGES CONDENSES VIDE OU INFXTS
1TANT !!'
RETURN

34 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOMBRE DE GROUPE
1 D'INDIVIDUS !!'
RETURN

35 WRITE(NW,*)'NOMBRE DE GROUPE D'INDIVIDUS INVALIDE !!'
RETURN

36 WRITE(NW,*)'PAS VALIDATION DU NOMBRE D'INDIVIDUS DANS
1 CHAQUE GROUPE !!'
RETURN

37 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DES NOMBRES


```

1 D'INDIVIDUS DANS LES GROUPES !!'
RETURN

38 WRITE(NW,*)'NOMBRE D'INDIVIDUS DANS UN GROUPE INVALIDE !!'
RETURN

39 WRITE(NW,*)'LA SOMME DES NOMBRES D'INDIVIDUS DANS LES
1 GROUPES NE VAUT PAS'
WRITE(NW,*)'LE NOMBRE TOTAL D'INDIVIDUS !!'
RETURN

40 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DE L'INDICATEUR DU
1 CHOIX DE LA'
WRITE(NW,*)'METHODE D'ARRET !!'
RETURN

41 WRITE(NW,*)'VALEUR DE L'INDICATEUR DU CHOIX DE LA METHODE
1 D'ARRET INVALIDE !!'
RETURN

42 WRITE(NW,*)'PAS VALIDATION DU NOMBRE DE VARIABLES A
1 SELECTIONNER OU DE LA VALEUR'
WRITE(NW,*)'FIXANT LA CROISSANCE MINIMALE DU
1 COEFFICIENT DE TSCHUPROW MULTIPLE !!'
RETURN

43 WRITE(NW,*)'PAS VALIDATION DU NOMBRE DE VARIABLES A
1 SELECTIONNER !!'
RETURN

44 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOMBRE DE VARI-
1 ABLES A SELECTIONNER !!'
RETURN

45 WRITE(NW,*)'NOMBRE DE VARIABLES A SELECTIONNER INVALIDE !!'
RETURN

46 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DE LA QUANTITE
1 FIXANT LA CROISSANCE'
WRITE(NW,*)'MINIMALE POUR LE COEFFICIENT DE TSCHUPROW
1 MULTIPLE !!'
RETURN

47 WRITE(NW,*)'QUANTITE FIXANT LA CROISSANCE MINIMALE POUR
1 LE COEFFICIENT'
WRITE(NW,*)'DE TSCHUPROW MULTIPLE INVALIDE !!'
RETURN

48 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU POIDS MAXIMUM
1 AUTORISE POUR'
WRITE(NW,*)'LA MODALITE NON-REPONDU D'UNE VARIABLE !!'
RETURN

49 WRITE(NW,*)'POIDS MAXIMUM AUTORISE POUR LA MODALITE NON
1-REPONDU'
WRITE(NW,*)'D'UNE VARIABLE INVALIDE !!'
RETURN

50 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU POIDS MINIMUM
1 POUR QU'UNE MODALITE'

```

```

WRITE(NW,*)'SOTT CONSERVEE !!'
RETURN

51  WRITE(NW,*)'POIDS MINIMUM POUR QUE UNE MODALITE SOTT CONSER
    1VFE INVALIDE !!'
    RETURN

52  WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DE L'INDICATEUR
    1 DU MOMENT DU CHOIX'
    WRITE(NW,*)'DE LA METHODE DE NETTOYAGE !!'
    RETURN

53  WRITE(NW,*)'INDICATEUR DU MOMENT DU CHOIX DE LA METHODE
    1 DE NETTOYAGE INVALIDE !!'
    RETURN

54  WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU CHOIX DE LA
    1 METHODE DE NETTOYAGE !!'
    RETURN

55  WRITE(NW,*)'CHOIX DE METHODE DE NETTOYAGE INVALIDE !!'
    RETURN

56  WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU SEUL
    1 DE REDONDANCE !!'
    RETURN

57  WRITE(NW,*)'SEUIL DE REDONDANCE INVALIDE !!'
    RETURN

58  WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOM DU FICHER'
    WRITE(NW,*)'DES CODAGES CONDENSES APRES SELECTION !!'
    RETURN

59  WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOM DU FICHER'
    WRITE(NW,*)'DES DIRECLES APRES NETTOYAGE !!'
    RETURN

60  WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOM DU FICHER
    1 CONTENANT LE'
    WRITE(NW,*)'TABLEAU X'DOX APRES LA SELECTION !!'
    RETURN

61  WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOM DU FICHER'
    WRITE(NW,*)'DES DONNEES POUR LE SECOND PROGRAMME !!'
    RETURN

62  WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DE L'INDICATEUR
    1 DU MODE'
    WRITE(NW,*)'DE SORTIE DES RESULTATS !!'
    RETURN

63  WRITE(NW,*)'INDICATEUR DU MODE DE SORTIE DES RESULTATS
    1 INVALIDE !!'
    RETURN

64  WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOM DU FICHER
    1 DES RESULTATS !!'
    RETURN

```


65 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOM DU FICHIER
1 CONTENANT '
WRITE(NW,*)'LE TABLEAU Y'DOY APRES LA SELECTION !!'
RETURN

66 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOM DU
1 FICHIER DES DONNEES !!'
RETURN

67 WRITE(NW,*)'FICHIER DE DONNEES VIDE OU INEXISTANT !!'
RETURN

68 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DES POSITIONS
1 DES MODALITES!!'
RETURN

69 WRITE(NW,*)'NOMBRE TOTAL DE MODALITES TROP IMPORTANT
1 POUR LES DIMENSIONS'
WRITE(NW,*)'DES TABLEAUX DANS CE PROGRAMME.'
WRITE(NW,*)'REVUEZ LA SELECTION OU CHANGEZ LES BORNES !!'
RETURN

70 WRITE(NW,*)'POSITIONS DE MODALITES INVALIDES !!'
RETURN

71 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DE L'INDICATEUR
1 D'EDITION DES POIDS'
WRITE(NW,*)'DES MODALITES APRES NETTOYAGE !!'
RETURN

72 WRITE(NW,*)'INDICATEUR D'EDITION DES POIDS DES MODALITES
1 APRES '
WRITE(NW,*)'NETTOYAGE INVALIDE !!'
RETURN

73 WRITE(NW,*)'ERREUR LORS DE LA LECTURE DE L'INERTIE MINIMALE
1 POUR UN FACTEUR Z !!'
RETURN

74 WRITE(NW,*)'INERTIE MINIMALE INVALIDE !!'
RETURN

75 WRITE(NW,*)'ERREUR LORS DE LA LECTURE DU POUVOIR
1 DISCRIMINANT CUMULE'
WRITE(NW,*)'A ATTEINDRE PAR LES FACTEURS Z !!'
RETURN

76 WRITE(NW,*)'POUVOIR DISCRIMINANT CUMULE INVALIDE !!'
RETURN

77 RETURN

78 RETURN

79 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DE L'INDICATEUR
1 D'EDITION DU '
WRITE(NW,*)'TABLEAU DES COEFFICIENTS DE TSCHUPROW DU PREMIER
1 ORDRE'
WRITE(NW,*)'ENTRE LES VARIABLES !!'
RETURN

R0 WRITE(NW,*)'INDICATEUR D'ÉDITION DU TABLEAU DES COEFFICIENTS
 1 DE TSCUPROW'
 WRITE(NW,*)'DU PREMIER ORDRE ENTRE LES VARIABLES INVALIDE !!'
 RETURN

 R1 WRITE(NW,*)'LE FICHIER CONTENANT LE TABLEAU Y'DOX EST VIDE
 1 OU INEXISTANT !!'
 RETURN

 R2 WRITE(NW,*)'LE FICHIER CONTENANT LE TABLEAU Y'DOX EST VIDE
 1 OU INEXISTANT !!'
 RETURN

 R3 WRITE(NW,*)'PROBLÈME LORS DE LA LECTURE DE LA FREQUENCE
 1 D'ÉDITION DU TABLEAU'
 WRITE(NW,*)'DES COEFFICIENTS DE TSCUPROW ENTRE LES
 1 VARIABLES !!'
 RETURN

 R4 WRITE(NW,*)'FREQUENCE D'ÉDITION DU TABLEAU DES COEFFICIENTS
 1 DE TSCUPROW'
 WRITE(NW,*)'ENTRE LES VARIABLES INVALIDE !!'
 RETURN

 R5 WRITE(NW,*)'PROBLÈME LORS DE LA LECTURE DE L'INDICATEUR
 1 D'ÉDITION DU VECTEUR'
 WRITE(NW,*)'DES COEFFICIENTS DE TSCUPROW DU PREMIER ORDRE
 1 ENTRE LES'
 WRITE(NW,*)'VARIABLES ET Y !!'
 RETURN

 R6 WRITE(NW,*)'VALEUR DE L'INDICATEUR D'ÉDITION DU VECTEUR DES
 1 COEFFICIENTS DE'
 WRITE(NW,*)'TSCUPROW DU PREMIER ORDRE ENTRE LES VARIABLES
 1 ET Y INVALIDE !!'
 RETURN

 R7 WRITE(NW,*)'PROBLÈME LORS DE LA LECTURE DE LA FREQUENCE
 1 D'ÉDITION DU VECTEUR'
 WRITE(NW,*)'DES COEFFICIENTS DE TSCUPROW ENTRE LES VARIABLES
 1 ET Y !!'
 RETURN

 R8 WRITE(NW,*)'FREQUENCE D'ÉDITION DU VECTEUR DES COEFFICIENTS
 1 DE TSCUPROW'
 WRITE(NW,*)'ENTRE LES VARIABLES ET Y INVALIDE !!'
 RETURN

 R9 WRITE(NW,*)'PROBLÈME LORS DE LA LECTURE DE L'INDICATEUR
 1 D'ÉDITION DU'
 WRITE(NW,*)'COEFFICIENT DE TSCUPROW MAXIMUM A CHAQUE
 1 ÉTAPE'
 RETURN

 R0 WRITE(NW,*)'VALEUR DE L'INDICATEUR D'ÉDITION DU COEFFICIENT
 1 DE TSCUPROW'
 WRITE(NW,*)'MAXIMUM A CHAQUE ÉTAPE INVALIDE !!'
 RETURN

91 WRITE(NW,*)'FICHER DE RESULTATS VIDE OU INEXISTANT !!'
RETURN

92 RETURN

93 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DE L'INDICATEUR
1 D'EDITION DU COEFFICIENT'
WRITE(NW,*)'DE TSCHUPROW MULTIPLE ET DE SA CROISSANCE A
1 CHAQUE ETAPE !!'
RETURN

94 WRITE(NW,*)'VALEUR DE L'INDICATEUR D'EDITION DU COEFFICIENT
1 DE TSCHUPROW'
WRITE(NW,*)'MULTIPLE ET DE SA CROISSANCE A CHAQUE ETAPE
1 INVALIDE !!'
RETURN

95 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU NOMBRE D'INDIVI
1 US A CLASSER !!'
RETURN

96 WRITE(NW,*)'NOMBRE D'INDIVIDUS A CLASSER INVALIDE !!'
RETURN

97 WRITE(NW,*)'PAS VALIDATION DES POIDS DES INDIVIDUS !!'
RETURN

98 WRITE(NW,*)'ERREUR LORS DE LA LECTURE DES POIDS DES
1 INDIVIDUS !!'
RETURN

99 WRITE(NW,*)'POIDS D'INDIVIDUS INVALIDES !!'
RETURN

100 WRITE(NW,*)'LA SOMME DES POIDS DES INDIVIDUS NE VAUT PAS
1 UN !!'
RETURN

101 WRITE(NW,*)'PAS VALIDATION DU CONTENU DU FICHER DES
1 LIBELLES !!'
RETURN

102 WRITE(NW,*)'ERREUR LORS DE LA LECTURE DE LIBELLES DE
1 VARIABLES !!'
RETURN

103 WRITE(NW,*)'ERREUR LORS DE LA LECTURE DE NOMBRE DE
1 MODALITES !!'
RETURN

104 WRITE(NW,*)'NOMBRE DE MODALITES INVALIDE !!'
RETURN

105 WRITE(NW,*)'ERREUR LORS DE LA LECTURE DE LIBELLES DE
1 MODALITES !!'
RETURN

106 WRITE(NW,*)'ERREUR LORS DE LA LECTURE DE LIBELLES DE
1 VARIABLES ET DE NOMBRE'
WRITE(NW,*)'DE MODALITES !!'

RETURN

- 107 WRITE(NW,*)'PAS VALIDATION DU FICHIER DES CODAGES
'CONDENSES !!'
RETURN
- 108 WRITE(NW,*)'ERREUR LORS DE LA LECTURE DU CONTENU DU
'FICHIER DES CODAGES CONDENSES !!'
RETURN
- 109 WRITE(NW,*)'VALEUR DE CODAGE INVALIDE !!'
RETURN
- 110 WRITE(NW,*)'ERREUR LORS DE LA LECTURE DE L'INDICATEUR
'1 DU TYPE'
WRITE(NW,*)'DE REGROUPEMENT !!'
RETURN
- 111 WRITE(NW,*)'VALEUR DE L'INDICATEUR DU TYPE DE REGROUPEMENT
'1 INVALIDE !!'
RETURN
- 112 WRITE(NW,*)'PAS VALIDATION DES MODALITES GROUPEES !!'
RETURN
- 113 WRITE(NW,*)'ERREUR LORS DE LA LECTURE DES MODALITES
'1 GROUPEES !!'
RETURN
- 114 WRITE(NW,*)'VALEUR DE MODALITE GROUPE INVALIDE !!'
RETURN
- 115 WRITE(NW,*)'NON RESPECT DE LA REGLE POUR LES MODALITES
'1 GROUPEES !!'
RETURN
- 116 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DE VOTRE CHOIX !!'
RETURN
- 117 WRITE(NW,*)'CHOIX INVALIDE !!'
RETURN
- 118 WRITE(NW,*)'PROBLEME LORS DE LA LECTURE DU CHOIX DE LA
'1 METHODE A '
WRITE(NW,*)'EMPLOYER POUR CETTE MODALITE !!'
RETURN
- 119 WRITE(NW,*)'METHODE A EMPLOYER POUR CETTE MODALITE
'1 INVALIDE !!'
RETURN
- 120 WRITE(NW,*)'PROBLEME DANS LE FICHIER DE DONNEES !!'
RETURN
- 121 WRITE(NW,*)'PROBLEME DANS LE FICHIER DES LIBELLES.'
WRITE(NW,*)'ARRET DE LA SAISIE DE CELUI-CI !!'
RETURN
- 122 WRITE(NW,*)'POUR QUE LES MODALITES GROUPEES SOIENT LES MEMES'
WRITE(NW,*)'POUR CHAQUE VARIABLE, IL FAUT AVOIR SPECIFIE'


```
WRITE(NW,*)'QUE LES VARIABLES ONT LE MEME NOMBRE DE MODALITES  
1 !!'  
RETURN
```

```
123 WRITE(NW,*)'ERREUR LORS DE LA LECTURE DE FRUPTA !!'  
RETURN
```

```
END
```

```
C+++++
```

```
C AFFICHAGE DE MESSAGES D'INTRODUCTION.  
C =====
```

```
SUBROUTINE MSGT(NW,NUM)
```

```
WRITE(NW,*)
```

```
C BRANCHEMENT VERS LE MESSAGE DE NUMERO NUM
```

```
GO TO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17  
1,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34  
2,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51  
3,52),NUM
```

```
C LISTE DES MESSAGES.
```

```
1 WRITE(NW, '(//)')  
WRITE(NW,*)  
WRITE(NW,*) *****  
WRITE(NW,*) * METHODE DE SAPORTA *  
WRITE(NW,*) * PROGRAMME NUMERO 1 *  
WRITE(NW,*) *****  
WRITE(NW, '(//)')  
RETURN
```

```
2 WRITE(NW,*)'SI VOUS INTRODUISEZ LES DONNEES AU  
1 TERMINAL, TAPÉZ 1. SINON TAPÉZ 0'  
RETURN
```

```
3 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DES DONNEES ?'  
RETURN
```

```
4 WRITE(NW,*)'QUEL EST LE NOMBRE DE VARIABLES EXPLICATIVES  
1 A CONSIDERER ?'  
RETURN
```

```
5 WRITE(NW,*)'SI LES VARIABLES SE SUIVENT DANS LE  
1 FICHIER DES CODAGES CONDENSES. '  
WRITE(NW,*)'TAPÉZ 1. SINON TAPÉZ 0'  
RETURN
```

```
6 WRITE(NW,*)'QUELS SONT LES NUMEROS DES VARIABLES A  
1 CONSIDERER ?'  
RETURN
```

```
7 WRITE(NW,*)'QUEL EST LE NOMBRE D'INDIVIDUS ?'  
RETURN
```

```
8 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DES POIDS DES  
1 INDIVIDUS ?'
```

RETURN

9 WRITE(NW,*)'S'IL EXISTE UN FICHIER DE LIBELLES
1. TAPPEZ 1. SINON TAPPEZ 0.'
RETURN

10 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DES LIBELLES ?'
RETURN

11 WRITE(NW,*)'SI TOUTES LES VARIABLES ONT LES MEMES
1 LIBELLES. TAPPEZ 1. SINON TAPPEZ 0.'
RETURN

12 WRITE(NW,*)'SI TOUTES LES VARIABLES ONT LE MEME
1 NOMBRE DE MODALITES.'
WRITE(NW,*)'TAPPEZ 1. SINON TAPPEZ 0.'
RETURN

13 WRITE(NW,*)'QUEL EST LE NOMBRE DE MODALITES ?'
RETURN

14 WRITE(NW,*)'QUELS SONT LES NOMBRES DE MODALITES DES
1 VARIABLES ?'
RETURN

15 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DES CODAGES
1 CONDENSEES ?'
RETURN

16 WRITE(NW,*)'QUEL EST LE NOMBRE DE GROUPE D'INDIVIDUS ?'
RETURN

17 WRITE(NW,*)'QUEL EST LE NOMBRE D'INDIVIDUS DANS CHAQUE
1 GROUPE ?'
RETURN

18 WRITE(NW,*)'SI VOUS CHOISISSEZ LA METHODE D'ARRET PAR
1 COEFFICIENT DE TSCHUPROW'
WRITE(NW,*)'MULTIPLE. TAPPEZ 1. DANS LE CAS CONTRAIRE
1. TAPPEZ 0.'
RETURN

19 WRITE(NW,*)'QUEL EST LE NOMBRE DE VARIABLES A
1 SELECTIONNER ?'
RETURN

20 WRITE(NW,*)'QUELLE EST LA VALEUR FIXANT LA CROISSANCE
1 MINIMALE POUR LE'
WRITE(NW,*)'COEFFICIENT DE TSCHUPROW MULTIPLE ?'
RETURN

21 WRITE(NW,*)'QUEL EST LE POTS MAXIMUM AUTORISE POUR LA'
WRITE(NW,*)'MODALITE NON-REPONDU D'UNE VARIABLE ?'
RETURN

22 WRITE(NW,*)'QUEL EST LE POTS MINIMUM POUR QU'UNE
1 MODALITE SOIT CONSERVEE ?'
RETURN

23 WRITE(NW,*)'SI VOUS CHOISISSEZ MAINTENANT LA METHODE

1 POUR SUPPRIMER LES TROUS DANS
 WRITE(NW,*)'LE TABLEAU DES CODAGES CONDENSES, TAPÉZ 1
 1, SINON TAPÉZ 0'
 RETURN

24 WRITE(NW,*)'QUELLE METHODE CHOISISSEZ-VOUS ?'
 WRITE(NW,*)' SUPPRIMER L'INDIVIDU ----> 1'
 WRITE(NW,*)' VENTILER AU HASARD ----> 2'
 WRITE(NW,*)' REGROUPER DES MODALITES ----> 3'
 RETURN

25 WRITE(NW,*)'QUEL EST LE SEUIL A PARTIR DUQUEL ON
 1 CONSIDERE QUE DEUX'
 WRITE(NW,*)'VARIABLES SONT REDONDANTES ?'
 RETURN

26 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DES CODAGES
 1 CONDENSES APRES SELECTION ?'
 RETURN

27 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DES ETIQUETTES
 1 APRES SELECTION ?'
 RETURN

28 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DEVANT
 1 CONTENIR LE TABLEAU DE BURT'
 WRITE(NW,*)'CROISANT LES MODALITES DES VARIABLES
 1 EXPLICATIVES'
 WRITE(NW,*)'APRES LA SELECTION ?'
 RETURN

29 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DEVANT
 1 CONTENIR LES DONNEES'
 WRITE(NW,*)'POUR LE SECOND PROGRAMME ?'
 RETURN

30 WRITE(NW,*)'QUEL EST LE MODE DE SORTIE POUR LES
 1 RESULTATS INTERMEDIATRES ?'
 WRITE(NW,*)' TERMINAL --> 1'
 WRITE(NW,*)' IMPRIMANTE --> 2'
 WRITE(NW,*)' FICHIER --> 3'
 RETURN

31 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DES RESULTATS
 1 INTERMEDIATRES ?'
 RETURN

32 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DEVANT
 1 CONTENIR LE TABLEAU'
 WRITE(NW,*)'DE BURT CROISANT LES MODALITES DES
 1 VARIABLES EXPLICATIVES'
 WRITE(NW,*)'AVEC CELLES DE Y APRES LA SELECTION ?'
 RETURN

33 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DES DONNEES ?'
 RETURN

34 WRITE(NW,*)'(/(/(/(/(')
 WRITE(NW,*)' *****'
 WRITE(NW,*)' * METHODE DE SAPOSTA. *'

```

WRITE(NW,*)
WRITE(NW,*)
WRITE(NW,“(//)”)
RETURN
* PROGRAMME NUMERO 2. *
*****

```

- 35 WRITE(NW,*)“SI VOUS VOULEZ LA LISTE DES POIDS DES
1 MODALITES APRES NETTOYAGE.”
WRITE(NW,*)“TAPEZ 1. SINON TAPEZ 0.”
RETURN
- 36 WRITE(NW,*)“QUELLE EST L'INERTIE MINIMALE POUR
1 UN FACTEUR Z ?”
RETURN
- 37 WRITE(NW,*)“QUEL EST LE POUVOIR DISCRIMINANT CUMULE
1 A ATTEINDRE.”
WRITE(NW,*)“POUR LES FACTEURS Z ?”
RETURN
- 38 WRITE(NW,*)“SI VOUS VOULEZ LE TABLEAU DES CODAGES
1 CONDENSES AVANT LA SELECTION.”
WRITE(NW,*)“TAPEZ 1. SINON TAPEZ 0.”
RETURN
- 39 WRITE(NW,*)“SI VOUS VOULEZ LE TABLEAU DES COEFFICIENTS
1 DE TSCHUPROW DU PREMIER
WRITE(NW,*)“ORDRE. TAPEZ 1. SINON TAPEZ 0.”
RETURN
- 40 WRITE(NW,*)“QUEL EST LE NOMBRE D'INDIVIDUS SUPPLEMENTAIRES
1 A CLASSER ?”
RETURN
- 41 WRITE(NW,*)“SI VOUS VOULEZ LES TABLEAUX DES
1 COEFFICIENTS DE TSCHUPROW
WRITE(NW,*)“PERIODIQUEMENT. DONNEZ LA PERIODE S=0”
RETURN
- 42 WRITE(NW,*)“SI VOUS VOULEZ LE TABLEAU DES
1 COEFFICIENTS DE TSCHUPROW DU
WRITE(NW,*)“PREMIER ORDRE AVEC Y. TAPEZ 1. SINON
1 TAPEZ 0.”
RETURN
- 43 WRITE(NW,*)“SI VOUS VOULEZ LES TABLEAUX DES
1 COEFFICIENTS DE TSCHUPROW
WRITE(NW,*)“AVEC Y PERIODIQUEMENT. TAPEZ 1.
1 SINON TAPEZ 0.”
RETURN
- 44 WRITE(NW,*)“SI VOUS VOULEZ LA VALEUR DU COEFFICIENT
1 DE TSCHUPROW
WRITE(NW,*)“MAXIMUM AVEC Y A CHAQUE ETAPE. TAPEZ 1
1. SINON TAPEZ 0.”
RETURN
- 45 RETURN
- 46 WRITE(NW,*)“SI VOUS VOULEZ LA VALEUR DU COEFFICIENT
1 DE TSCHUPROW MULTIPLE”

WRITE(NW,*)'AVEC Y ET SA CROISSANCE A CHAQUE ETAPE
1. TAPPEZ 1 SINON TAPPEZ 0.'
RETURN

47 RETURN

48 WRITE(NW,*)'SI LE RECRUTEMENT EST LE MEME POUR
1 TOUTES LES VARIABLES.'
WRITE(NW,*)'TAPPEZ 1. SINON TAPPEZ 0.'
RETURN

49 WRITE(NW,*)'QUELLE EST LA LISTE DES MODALITES GROUPEES ?'
RETURN

50 WRITE(NW,*)'VOULEZ-VOUS VOIR LES POIDS ET LES NOMBRES
1 D'INDIVIDUS CONCERNES'
WRITE(NW,*)'POUR LES AUTRES MODALITES DE CETTE
1 VARIABLE ?'
WRITE(NW,*)'SI OUI, TAPPEZ 1; SI NON, TAPPEZ 0.'
RETURN

51 WRITE(NW,*)'SI VOUS VOULEZ JUMPER CETTE MODALITE
1 AVEC UNE AUTRE, TAPPEZ 1.'
WRITE(NW,*)'SI VOUS VOULEZ LAISSER TOMBER LES
1 INDIVIDUS S'Y RAPPORTANT, TAPPEZ 0.'
RETURN

52 WRITE(NW,*)'QUEL EST LE NOM DU FICHIER DEVANT CONTENIR
1 LE TABLEAU DE BURT'
WRITE(NW,*)'ORDONNANT LES MODALITES DES VARIABLES EXPLICATIVES'
WRITE(NW,*)'AVANT LA SELECTION ?'
RETURN

END

C+++++

```

C*****C
C*                                          *C
C*      METHODE DE SAFORTA, DEUXIEME PROGRAMME      *C
C*                                          *C
C*****C

```

```

C      Ce programme comporte l'analyse factorielle discriminante
C      sur les variables retenues, le reclassement des individus
C      de depart et le classement des individus supplementaires.

```

```

C      Pour l'analyse factorielle, cfr. memoire chapitre II et IV
C      Pour le classement des individus, cfr. idem.

```

```

C=====

```

```

      PROGRAM PRINC2

```

```

C      declarations
C      -----

```

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Y)
      IMPLICIT LOGICAL (Z)

```

```

      CHARACTER*(10) FDON,FBURT,FBURTY,FRES,FCOD

```

```

      DIMENSION MPDS(101),A(150,150),PM(150),PMDMI(150)
      1,VLP(134),VP(150,134),NUIGR(9),PI(4000),YTDPY(9)
      2,VTR(150),YTDPIZ(9,150),NTRI(134),ROT(150),VP1(150)
      3,NBPG(9),NCODI(75),VALEUR(9)

```

```

      DATA NDIM/150/
      DATA NUMTER,NUMLPT,NBURT,NBURTY,NFDON,NUMFR,NUMCC
      1/5,3,50,51,52,53,54/
      DATA NUMIN,NUMAX/1,100/
      DATA NIMIN,NIMAX/2,4000/
      DATA NICLMI/0/
      DATA PIINF,PISUP,EPSPi,SOMPI,EPSSPI/0.00,1.00,1.D-15
      1,1.00,1.D-15/
      DATA NGRMIN,NGRMAX/2,9/
      DATA VLPINF,VLPSP,EPVLP/0.00,1.00,1.D-15/
      DATA PDCINF,EPSPDC/0.00,1.D-15/
      DATA NOUT1,NOUT2,NOUT3/1,2,3/

```

```

      CALL MSBT(NUMTER,34)
      CALL SAVAS(NUMTER,NUMTER,NUMTER,FDON,33,66,67,ZF)

```

```

C      lecture des donnees.

```

```

      OPEN (UNIT=NFDON,FILE=FDON,DEVICE='DSK')
      CALL SAVAS2(NUMTER,NFDON,NUMTER,NU,NUMIN,NUMAX,0,5,6,ZE)
      WRITE(NUMTER,*)NU
      IF (ZE) STOP
      CALL SAVAS13(NUMTER,NFDON,NUMTER,MPDS,NU+1,NU+1,0,150
      1,0,68,69,70,ZE)
      IF (ZE) STOP
      CALL SAVAS2(NUMTER,NFDON,NUMTER,NI,NIMIN,NIMAX,0,14,15,ZE)
      IF (ZE) STOP
      CALL SAVAS4(NUMTER,NFDON,NUMTER,NICLA,NICLMI,0,95,96,ZE)
      IF (ZE) STOP
      CALL SAVAS14(NUMTER,NFDON,NUMTER,PI,NI,NIMAX,PIINF,PISUP

```



```

1,EPSP1,SOMPI,EPSSPI,0,98,99,100,ZE)
IF (ZE) STOP
CALL SAVA2(NUMTER,NFDON,NUMTER,NGR,NGRMIN,NGRMAX
1,0,34,35,ZE)
IF (ZE) STOP
CALL SAVA10(NUMTER,NFDON,NUMTER,NVIGR,NGR,NGRMAX,1
1,NI-1,NI,0,37,38,39,ZE)
IF (ZE) STOP
CALL SAVA5(NUMTER,NFDON,NUMTER,FBURT,0,60,81,ZE)
IF (ZE) STOP
CALL SAVA5(NUMTER,NFDON,NUMTER,FBURTY,0,65,82,ZE)
IF (ZE) STOP
CALL SAVA5(NUMTER,NFDON,NUMTER,FCOD,0,32,33,ZE)
IF (ZE) STOP
CALL SAVA3(NUMTER,NFDON,NUMTER,NOUT,NOUT1,NOUT2,NOUT3
1,0,62,63,ZE)
IF (ZE) STOP
IF (NOUT.EQ.NOUT3) THEN
    CALL SAVA5(NUMTER,NFDON,NUMTER,FRES,0,64,91,ZE)
    IF (ZE) STOP
END IF
CALL SAVA6(NUMTER,NFDON,NUMTER,VLPMIN,VLPINF,VLPSUP
1,EPVLP,0,73,74,ZE)
IF (ZE) STOP
CALL SAVA6(NUMTER,NFDON,NUMTER,PDISCU,PDCINF,DFLOAT(NGR-1)
1+1,D-8,EPSPDC,0,75,76,ZE)
IF (ZE) STOP
CLOSE(NFDON)

```

C affectation de NUMRES.

```

IF (NOUT.EQ.NOUT3) THEN
    OPEN (UNIT=NUMFR,FILE=FRES,DEVICE='DSK',ACCESS='APPEND')
    NUMRES=NUMFR
ELSE
    IF (NOUT.EQ.NOUT1) THEN
        NUMRES=NUMTER
    ELSE
        NUMRES=NUMLPT
    END IF
END IF
NTOTM=MPOS(NV+1)

WRITE(NUMRES,*)
WRITE(NUMRES,*)'ANALYSE FACTORIELLE DISCRIMINANTE :/'
WRITE(NUMRES,*)'=====/'
WRITE(NUMRES,*)

```

C Lecture du tableau de burt croisant les modalites des
C variables explicatives

```

OPEN (UNIT=NBURT,FILE=FBURT,DEVICE='DSK')
DO 100 I=1,NTOTM
    READ(NBURT,*)(A(I,J),J=1,NTOTM)
100 CONTINUE
CLOSE(NBURT)

```

C formation du vecteur des poids des modalites et de leur
C racines

```

DO 110 I=1,NTOTM
    PM(I)=A(I,I)
    PMDEMI(I)=SQRT(PM(I))
110 CONTINUE

C     pre et post multiplication du tableau de burt par la matrice
C     diagonale des inverses des racines carrees des poids des
C     modelites

A(1,1)=1.D0
DO 120 I=2,NTOTM
    DO 130 J=1,I-1
        A(I,J)=(A(I,J)/PMDEMI(I))/PMDEMI(J)
        A(J,I)=A(I,J)
130 CONTINUE
    A(I,I)=1.D0
120 CONTINUE

C     recherche des valeurs propres et des vecteurs propres de la
C     matrice transformee.

UNV=DFLOAT(NV)+1.D-5
CALL ANACOR(NDIM,NTOTM,A,NBVLP,NTOTM-NV+1,VLP,VP,VLPMIN,UNV
1,IERR,NUMTER)

IF (IERR.NE.0) THEN
    WRITE(NUMTER,*)
    WRITE(NUMTER,*)'PB LORS DE L'ANALYSE DES CORRESPONDANCES'
    STOP
END IF

C     transformation des vecteurs propres trouves pour avoir ceux
C     de la matrice initiale.

DO 140 I=1,NTOTM
    DO 140 J=1,NBVLP
        VP(I,J)=VP(I,J)/PMDEMI(I)
140 CONTINUE

C     rotation des vecteurs propres pour que celui correspondant
C     a la plus grande valeur propre ne contienne que des 1 a une
C     constante pres, et que ainsi les autres soient centres.

XNORM=0.D0
DO 8000 I=1,NTOTM
    ROT(I)=1.D0/SQRT(DFLOAT(NV))-VP(I,NBVLP)
    XNORM=XNORM+ROT(I)*ROT(I)*PM(I)
8000 CONTINUE

IF (XNORM.GT.1.D-12) THEN
    DO 8010 K1=1,NBVLP
        DO 8020 K2=1,NTOTM
            RES=0.D0
            DO 8030 K3=1,NTOTM
                IF (K3.EQ.K2) THEN
                    RES=RES+VP(K3,K1)-2.D0*ROT(K2)*ROT(K3)*
1 PM(K3)*VP(K3,K1)/XNORM
                ELSE
                    RES=RES-2.D0*ROT(K2)*ROT(K3)*PM(K3)*VP(K3,K1)
1 /XNORM
            END IF
        END IF
    END IF

```



```

8030      CONTINUE
          IF (ABS(RES).GT.1.D-8) THEN
              VP1(K2)=RES
          ELSE
              VP1(K2)=0.D0
          END IF
8020      CONTINUE
          DO 8040 K4=1,NTOTM
              VP(K4,K1)=VP1(K4)
8040      CONTINUE
8010      CONTINUE
END IF

C      normalisation des codages descriptifs a l'inverse de la
C      racine carree de la valeur propre correspondante.

      DO 150 I=1,NBVLP
          XNORM=0.
          DO 160 J=1,NTOTM
              XNORM=XNORM+VP(J,I)*VP(J,I)*PM(J)
160      CONTINUE
          DO 170 J=1,NTOTM
              VP(J,I)=VP(J,I)/SQRT(VLP(I)*XNORM)
170      CONTINUE
150      CONTINUE

C      impression des codages descriptifs et des inerties des
C      facteurs z.

      WRITE(NUMRES,*)
      WRITE(NUMRES,*)'CODAGES DESCRIPTIFS ET INERTIES
1 CORRESPONDANTES'
      WRITE(NUMRES,*)'-----'
      1-----'
      DO 180 I=1,NBVLP
          WRITE(NUMRES,*)
          WRITE(NUMRES,*)'CODAGE DESCRIPTIF NUMERO :',I
          WRITE(NUMRES,*)'-----'
          WRITE(NUMRES,*)
          WRITE(NUMRES,*(X,150(X,D14,7))')(VP(J,I),J=1,NTOTM)
          WRITE(NUMRES,*)
          WRITE(NUMRES,*)'INERTIE EXPLIQUEE :',VLP(I)
180      CONTINUE

C      calcul des poids des groupes.

      I=1
      NBO=0
      NBTOT=1
      SOM=0.
      DO 190 WHILE (I.LE.NGR)
          IF ((NBTOT-NBO).LE.NVIGR(I)) THEN
              SOM=SOM+PI(NBTOT)
              NBTOT=NBTOT+1
          ELSE
              YIDPY(I)=SOM
              SOM=0.
              I=I+1
              NBO=NBTOT-1
          END IF
190      CONTINUE

```



```

C      formation de la matrice YTFPZ
      NFNT=NBULP-1

      OPEN (UNIT=NBURTY,FILE=FBURTY,DEVICE='DSK')
      DO 200 I=1,NGR
        READ(NBURTY,*)(VTR(L),L=1,NTOTM)
        DO 210 K=1,NBULP
          SOM=0.
          DO 220 J=1,NTOTM
            SOM=SOM+VTR(J)*VP(J,K)
220          CONTINUE
            YTFPZ(I,K)=SOM
210          CONTINUE
200        CONTINUE
      CLOSE(NBURTY)

C      calcul des pouvoirs discriminants de facteurs z.

      DO 230 I=1,NBULP
        SOM=0.
        DO 240 J=1,NGR
          SOM=SOM+YTFPZ(J,I)*YTFPZ(J,I)/YTFPY(J)
240        CONTINUE
          VTR(I)=SOM
          A(I,I)=SOM
230        CONTINUE

C      impression des pouvoirs discriminants

      WRITE(NUMRES,*)
      WRITE(NUMRES,*)'POUVOIRS DISCRIMINANTS DES DIFFERENTS
1 FACTEURS : '
      WRITE(NUMRES,*)'-----
1-----'
      WRITE(NUMRES,*)
      WRITE(NUMRES,')(X,134(X,D14.7))')(VTR(I),I=1,NBULP)

C      tri par ordre de pouvoir discriminant decroissant jusqu'a
C      obtention d'une somme superieure ou egale a PDISCU.
C      impression des numeros des facteurs z conserves

      WRITE(NUMRES,*)
      WRITE(NUMRES,*)'FACTEURS Z CONSERVES : '
      WRITE(NUMRES,*)'-----'
      DO 250 I=1,NFNT
        NTRI(I)=0
250      CONTINUE
      SOM=0.
      NBGAR=0
      ZCONT=.TRUE.
      DO 260 WHILE (ZCONT)
        PDISM=0.
        IMAX=0
        DO 270 I=1,NFNT
          IF ((VTR(I).GE.PDISM).AND.(NTRI(I).EQ.0)) THEN
            PDISM=VTR(I)
            IMAX=I
          END IF
270        CONTINUE

```



```

SOM=SOM+PDISM
NBGAR=NBGAR+1
NTRI(IMAX)=NBGAR
WRITE(NUMRES,*)'FACTEUR ',IMAX
IF ((SOM.GE.PDISCU).OR.(NBGAR.EQ.NFNT)) ZCONT=.FALSE.
260 CONTINUE

```

C mise à jour à jour de VP, YTFPZ et de la diagonale de A.

```

DO 280 I=1,NBGAR
  DO 290 J=1,NBVLP
    IF (NTRI(J).EQ.I) THEN
      IF (J.NE.I) THEN
        NTRI(J)=NTRI(I)
        IF (NTRI(I).NE.0) THEN
          DO 300 K=1,NTOTM
            PMDEMI(K)=VP(K,I)
300 CONTINUE
            DO 310 K=1,NGR
              VTR(K)=YTFPZ(K,I)
310 CONTINUE
              AAUX=A(I,I)
            END IF
            DO 320 K=1,NTOTM
              VP(K,I)=VP(K,J)
320 CONTINUE
            DO 330 K=1,NGR
              YTFPZ(K,I)=YTFPZ(K,J)
330 CONTINUE
              A(I,I)=A(J,J)
            IF (NTRI(I).NE.0) THEN
              DO 340 K=1,NTOTM
                VP(K,J)=PMDEMI(K)
340 CONTINUE
                DO 350 K=1,NGR
                  YTFPZ(K,J)=VTR(K)
350 CONTINUE
                  A(J,J)=AAUX
                END IF
              END IF
            END IF
          CONTINUE
          NTRI(I)=0
280 CONTINUE

```

```

WRITE(NUMRES,*)
WRITE(NUMRES,*)'POUVOIR DISCRIMINANT CUMULE :',SOM

```

C calcul de la matrice $ZtDpY(YtDpY)^{-1}YtDpZ$

```

IF (NBGAR.GT.1) THEN
  DO 400 I=2,NBGAR
    DO 400 J=1,I-1
      SOM=0.DO
      DO 410 K=1,NGR
        SOM=SOM+YTFPZ(K,I)*YTFPZ(K,J)/YTFPY(K)
410 CONTINUE
        A(I,J)=SOM
        A(J,I)=SOM
400 CONTINUE
  END IF

```

C Analyse discriminante sur les facteurs z conserves

CALL ANDISZ(NDIM,NBGAR,A,VLP,IERR,NUMTER)

```
IF (IERR.NE.0) THEN
  WRITE(NUMTER,*)
  WRITE(NUMTER,*)'PROBLEME LORS DE L'ANALYSE DISCRIMI
INANTE SUR LES FACTEURS Z !!'
  IF (NUMRES.EQ.NUMFR) CLOSE(NUMRES)
  STOP
END IF
```

C impression des codages discriminants pour les facteurs z

```
WRITE(NUMRES,*)
WRITE(NUMRES,*)'CODAGES DISCRIMINANTS POUR LES FACTEURS Z'
WRITE(NUMRES,*)'-----'
ZCONT=.TRUE.
I=NBGAR
DO 420 WHILE (ZCONT)
  WRITE(NUMRES,*)
  WRITE(NUMRES,*)'CODAGE :',NBGAR-I+1
  WRITE(NUMRES,*)'-- -- --'
  WRITE(NUMRES,')(X,134(D14.7,X))')(A(J,I),J=1,NBGAR)
  WRITE(NUMRES,*)
  WRITE(NUMRES,*)' -> POUVOIR DISCRIMINANT : ',VLP(I)
  WRITE(NUMRES,*)
  IF (I.EQ.1) THEN
    ZCONT=.FALSE.
  ELSE
    IF (ABS(VLP(I-1)).LT.1.D-8) THEN
      ZCONT=.FALSE.
    ELSE
      I=I-1
    END IF
  END IF
420 CONTINUE
ID=I
```

C calcul et impression des codages discriminants correspondants

```
WRITE(NUMRES,*)
WRITE(NUMRES,*)'CODAGES DISCRIMINANTS RECHERCHES'
WRITE(NUMRES,*)'-----'
DO 430 I=NBGAR,ID,-1
  DO 440 J=1,NTOTM
    SOM=0.D0
    DO 450 K=1,NBGAR
      SOM=SOM+A(K,I)*VP(J,K)
450 CONTINUE
    VP1(J)=SOM
440 CONTINUE
    WRITE(NUMRES,*)
    WRITE(NUMRES,*)'CODAGE :',NBGAR-I+1
    WRITE(NUMRES,*)
    WRITE(NUMRES,')(X,150(D14.7,X))')(VP1(J),J=1,NTOTM)
430 CONTINUE
```

C formation de la matrice des centres de gravites des groupes


```

DO 460 I=1,NGR
DO 460 J=1,NBGAR
  YTDPZ(I,J)=YTDPZ(I,J)/YTDOPY(I)
460 CONTINUE

C      formation du vecteur des produits scalaires des centres de
C      gravites des facteurs z pour les differents groupes

DO 470 I=1,NGR
  SOM=0.DO
  DO 480 J=1,NBGAR
    SOM=SOM+YTDPZ(I,J)*YTDPZ(I,J)
480 CONTINUE
  YTDOPY(I)=SOM
470 CONTINUE

C      formation de la matrice 2 * (YtDeY)**-1 YtDeZ Ut
DO 490 I=1,NGR
  DO 500 J=1,NTOTM
    SOM=0.DO
    DO 510 K=1,NBGAR
      SOM=SOM+YTDPZ(I,K)*VP(J,K)
510 CONTINUE
    VTR(J)=2.DO*SOM
500 CONTINUE
  DO 520 J=1,NTOTM
    YTDPZ(I,J)=VTR(J)
520 CONTINUE
490 CONTINUE

C      reclassement des individus de depart

OPEN (UNIT=NUMCC,FILE=FCOD,DEVICE='DSK')
WRITE(NUMRES,*)
WRITE(NUMRES,*)'RECLASSEMENT DES INDIVIDUS DE DEPART :'
WRITE(NUMRES,*)'===== '
WRITE(NUMRES,*)
NBBC=0
NBT=0
DO 530 IGR=1,NGR
  WRITE(NUMRES,*)
  WRITE(NUMRES,*)'GROUPE : ',IGR
  WRITE(NUMRES,*)'- - - - '
  WRITE(NUMRES,*)
  WRITE(NUMRES,*)'  NOMBRE D'INDIVIDUS : ',NVIGR(IGR)
  WRITE(NUMRES,*)

  DO 540 I=1,NGR
    NBPG(I)=0
540 CONTINUE
  DO 550 I=1,NVIGR(IGR)
    READ(NUMCC,*)(NCODI(J),J=1,NV)
    DO 560 IV=1,NGR
      VALEUR(IV)=YTDOPY(IV)
560 CONTINUE
    DO 580 J=1,NV
      I1=MPOS(J)+NCODI(J)
      DO 570 IV=1,NGR
        VALEUR(IV)=VALEUR(IV)-YTDPZ(IV,I1)
570 CONTINUE
580 CONTINUE

```

```

      IMIN=IGR
      VMIN=VALEUR(IGR)
      DO 590 IV=1,NGR
        IF (IV.NE.IGR) THEN
          IF (VMIN.GT.VALEUR(IV)) THEN
            VMIN=VALEUR(IV)
            IMIN=IV
          END IF
        END IF
590      CONTINUE
      WRITE(NUMRES,*)'      INDIVIDU ',I,' -> GROUPE ',IMIN
      NBPG(IMIN)=NBPG(IMIN)+1
550    CONTINUE
      NBBC=NBBC+NBPG(IGR)
      NBT=NBT+NVIGR(IGR)
      WRITE(NUMRES,*)
      WRITE(NUMRES,*)'RESULTAT DE L''AFFECTATION : '
      WRITE(NUMRES,*)
      DO 600 IV=1,NGR
        WRITE(NUMRES,*)'GROUPE : ',IV,' -> ',NBPG(IV)
600      CONTINUE
      WRITE(NUMRES,*)
      WRITE(NUMRES,*)'% DE BIEN CLASSES : ',DFLOAT(NBPG(IGR))
      1/DFLOAT(NVIGR(IGR))*1.D2
530    CONTINUE
      WRITE(NUMRES,*)
      WRITE(NUMRES,*)'POURCENTAGE TOTAL DE BIEN-CLASSES : ',
      1DFLOAT(NBBC)/DFLOAT(NBT)*1.D2
      IF (NICLA.GT.0) THEN
        WRITE(NUMRES,*)
        WRITE(NUMRES,*)'CLASSEMENT DES INDIVIDUS SUPPLEMENTAIRES : '
        WRITE(NUMRES,*)'-----'
        DO 750 I=1,NICLA
          READ(NUMCC,*)(NCODI(J),J=1,NV)
          DO 760 IV=1,NGR
            VALEUR(IV)=YTDPY(IV)
760          CONTINUE
          DO 780 J=1,NV
            I1=MPOS(J)+NCODI(J)
            DO 770 IV=1,NGR
              VALEUR(IV)=VALEUR(IV)-YTFPZ(IV,I1)
770            CONTINUE
780          CONTINUE
          IMIN=1
          VMIN=VALEUR(1)
          DO 790 IV=2,NGR
            IF (VMIN.GT.VALEUR(IV)) THEN
              VMIN=VALEUR(IV)
              IMIN=IV
            END IF
790          CONTINUE
          WRITE(NUMRES,*)'      INDIVIDU ',I,' -> GROUPE ',IMIN
750        CONTINUE
      END IF
      CLOSE(NUMCC)
      IF (NUMRES.EQ.NUMFR) CLOSE(NUMRES)
      END

```



```

SUBROUTINE ANACOR(NM,N,A,M,MM,W,VP,PB,UB,IERR,NUMTER)

IMPLICIT DOUBLE PRECISION (A-H,O-Y)

DIMENSION A(NM,N),W(MM),VP(NM,MM),FV1(180),FV2(180)
1,FV3(180),FV4(180),FV5(180),FV6(180),FV7(180),FV8(180)
2,IV1(160)

CALL TRED1(NM,N,A,FV1,FV2,FV3)
EPS1=0.D0
CALL BISECT(N,EPS1,FV1,FV2,FV3,PB,UB,MM,M,W,IV1,IERR
1,FV4,FV5)
IF (IERR.EQ.0) THEN
    CALL TINVIT(NM,N,FV1,FV2,FV3,M,W,IV1,VP,IERR,FV4,FV5
1,FV6,FV7,FV8)
    IF (IERR.EQ.0) THEN
        CALL TRBAK1(NM,N,A,FV2,M,VP)
    ELSE
        WRITE(NUMTER,*)'PLUS QUE 5 ITERATIONS POUR DETERMINER '
        WRITE(NUMTER,*)'ENTRE AUTRES LE VECTEUR PROPRE '
1,-IERR
    END IF
ELSE
    WRITE(NUMTER,*)'PLUS DE VLP QUE PREVU DANS L''INTERVALLE !'
END IF
RETURN
END

```

SUBROUTINE ANDISZ(NM,N,A,W,IERR,NUMTER)

IMPLICIT DOUBLE PRECISION (A-H,O-Y)

DIMENSION A(NM,NM),W(NM),FV1(150)

CALL TRED2(NM,N,A,W,FV1,A)

CALL TQL2(NM,N,W,FV1,A,IERR)

IF (IERR.NE.0) THEN

WRITE(NUMTER,*)'PLUS QUE 30 ITERATIONS NECESSAIRES POUR
1 DETERMINER'

WRITE(NUMTER,*)'LA VALEUR PROPRE ',L

END IF

RETURN

END

@

REFERENCES.

=====

R1 : Saporta G.

"Une méthode et un programme d'analyse discriminante pas-à-pas sur variables qualitatives".

Colloques I.R.I.A. premières journées.

R2 : Romeder J.M.

"Méthodes et programmes d'analyse discriminante".

Dunod.

R3 : Bragard L., Richard J.F., Simar L.

"Analyse discriminante".

Recyclage en statistique. Comité de statistique U.C.L.

Séminaire de Mathématiques Appliquées aux Sciences Humaines,

Facultés Universitaires St Louis.

R4 : Lebart, Morineau A., Tabard N.

"Techniques de la description statistique".

Dunod.

R5 : Saporta G.

"Liaisons entre plusieurs ensembles de variables et codages de données qualitatives".

Thèse de 3^{ième} cycle. Université Paris VI. 1975.

R6 : Diday E., Lemaire J., Pouget J., Testu F.

"Elements d'analyse de données".

Dunod.

R7 : Lebart L.

"L'orientation du dépouillement de certaines enquêtes par l'analyse
des correspondances multiples".

Consommation n° 2, 1975.

R8 : B.J. Smith, J.M. Boyle, J.J. Dongarra, B.S. Garbow, Y. Ikebe,
V.C. Klema, C.B. Moler.

"Matrix Eigensystem Routines - EISPACK Guide". Second edition.

Springer - Verlag

R9 : J.H. Wilkinson.

"The Algebraic Eigenvalue Problem".

Clarendon Press - Oxford.

R10 : Dehedin.

Thèse de 3^{ième} cycle. Université de Paris VI. 1975.